



**UNIVERSIDAD NACIONAL DE INGENIERÍA  
RECINTO UNIVERSITARIO “SIMÓN BOLÍVAR”  
FACULTAD DE ELECTROTECNIA Y COMPUTACIÓN**

**TRABAJO MONOGRÁFICO**

**“Desarrollo de herramienta Informática de  
Inteligencia de Negocios para el Sistema Nacional  
de Inversión Pública de la República de Nicaragua”**

**PARA OPTAR AL TÍTULO DE  
*INGENIERO EN COMPUTACIÓN***

**ELABORADO POR:**

***Br. Judith del Carmen Navarro Ordeñana***

***Br. José Luis Zamora Osegueda***

**TUTOR:**

***Ing. José Leónidas Díaz Chow***

**MANAGUA, NICARAGUA**

**ABRIL 2016**

## DEDICATORIA

*Judith Navarro.*

*Primeramente a Dios por haberme permitido culminar con éxito esta etapa de mi vida, a mi familia que con mucho cariño me ha motivado para persistir en mis estudios.*

*A nuestro tutor el Ing. José Díaz Chow que siempre mostró su actitud positiva, dedicación y tiempo para guiarnos por el camino a la finalización de este trabajo Monográfico.*

*A todas las personas que me brindaron su apoyo en este largo y difícil camino.*

*A todos ellos, mi eterno agradecimiento!...*

*José Zamora*

*Esta monografía se la dedico a mis padres José Luis y Leda Rosa que me apoyaron en todo momento para culminar con éxito esta etapa de mi vida.*

## **RESUMEN**

El análisis de datos en las empresas es una necesidad cotidiana para guiar la toma de decisiones. Este proceso ha tenido significativos cambios en las últimas décadas gracias a la llegada de herramientas tecnológicas para el análisis de información y soporte a la toma de decisiones que conforman una disciplina denominada Inteligencia de Negocios. Hoy en día la inteligencia de negocios ha sido una de las herramientas más completas para la toma de decisiones empresarial, tanto a nivel gerencial como al nivel operativo. La evolución de esta tendencia está en la gestión de la información, el tiempo que se emplea en procesarla y en qué medida es utilizada eficientemente.

El presente proyecto surge como respuesta a la necesidad de dotar al Sistema Nacional de Inversión Pública (SNIP) de una herramienta estratégica de inteligencia de Negocios que facilite la gestión de los procesos de seguimiento y monitoreo de la ejecución del Programa de Inversión Pública. Para llevar a cabo dicho proyecto, se empleó una metodología adecuada de desarrollo de sistemas BI (metodología de Kimball) teniendo como recurso esencial y primario el diseño y desarrollo de un Data Warehouse, sobre el cual se cimenta un Sistema de Soporte a Decisiones y un Cuadro de Mando Integral.

El Data Warehouse se implementó con un abordaje relacional sobre el gestor de base de datos SQL Server, y la herramienta BI, como aplicación ASP.Net, aprovechando los componentes de BI de Radarsoft y de Charting de HighCharts.

La herramienta desarrollada se puso en producción y se capacitaron a cinco usuarios del nivel táctico para realizar pruebas y evaluar si la herramienta cumple con los objetivos propuestos. Los resultados de dicha evaluación, evidenciaron que la herramienta de Inteligencia de Negocios implementada, provee todos los beneficios esperados a los usuarios, lográndose así los objetivos del proyecto exitosamente.

# ÍNDICE

1	INTRODUCCIÓN .....	1
2	OBJETIVOS .....	2
2.1	Objetivo General .....	2
2.2	Objetivos Específicos.....	2
3	JUSTIFICACIÓN.....	3
4	MARCO TEÓRICO.....	5
4.1	Inteligencia de Negocios .....	5
4.2	Niveles de toma de decisión en una organización .....	6
4.3	Data WareHouse o Almacén de Datos .....	7
4.3.1	DataMart .....	7
4.3.2	Modelado dimensional del Data Warehouse.....	8
4.3.3	Tipos de Modelo Dimensional.....	9
4.3.4	Modelo en Estrella .....	9
4.3.5	Modelo en Copo de Nieve.....	10
4.3.6	ETL (Extracción, Transformación y Carga).....	11
4.4	Tipos de procesamiento de datos .....	11
4.4.1	OLTP: On Line Transactional Processing .....	11
4.4.2	OLAP: On Line Analytical Processing .....	12
4.5	Dashboard (Panel de Indicadores Gerenciales) .....	12
4.6	Metodologías de desarrollo de Sistemas de Inteligencia de Negocios.....	13
4.7	Proceso de desarrollo de Software .....	16
5	ANÁLISIS Y PRESENTACIÓN DE RESULTADOS.....	18
5.1	Metodología Empleada .....	18
5.2	Definición de roles del proyecto según SCRUM.....	18
5.3	Desarrollo de la Solución.....	19
5.3.1	Planificación del proyecto .....	19
5.3.2	Definición de los requerimientos del negocio .....	19
5.3.3	Definición del Backlog inicial de SCUM .....	21
5.3.4	Modelado dimensional.....	22
5.3.5	Diseño físico .....	26
5.3.6	Diseño y desarrollo de la presentación de datos .....	28

5.3.7	Diseño de la arquitectura técnica .....	29
5.3.8	Selección de productos e instalación .....	30
5.3.9	Especificación de aplicaciones para usuarios finales .....	32
5.3.10	Desarrollo de aplicaciones para usuarios finales .....	34
5.3.11	Mantenimiento y crecimiento.....	39
5.3.12	Implantación de la solución .....	39
5.3.13	Resultados de la aplicación de la solución .....	41
5.3.14	Resultados en la bitácora del Backlog.....	42
5.4	Evaluación de los resultados .....	46
6	CONCLUSIONES Y RECOMENDACIONES.....	53
7	BIBLIOGRAFÍA.....	56
8	ANEXOS .....	58
8.1	Anexo A: Manual de referencia técnica de la Herramienta .....	58
8.1.1	Instalación de Suite de controles OLAP de Radarsoft .....	58
8.1.2	Desarrollo de la Herramienta de BI.....	60
8.1.3	Procedimiento para llenar el RadarCube .....	67
8.1.4	Procedimiento para generar gráficos con HighCharts .....	79

## Índice de Tablas

<i>Tabla 1: Niveles de Toma de Decisiones</i> .....	6
<i>Tabla 2: Metodologías de Diseño de DataWareHouse</i> .....	13
<i>Tabla 3: Personas y roles del proyecto</i> .....	18
<i>Tabla 4: Tareas Programadas en el Backlog inicial</i> .....	21

## Índice de Ilustraciones

<i>Ilustración 1: Esquema en Estrella</i> .....	10
<i>Ilustración 2: Esquema en Copo de Nieve</i> .....	10
<i>Ilustración 3: Ciclo de vida del sistema BI, Metodología de Kimball.</i> .....	16
<i>Ilustración 4: Flujo de Trabajo de la Metodología Scrum.</i> .....	17
<i>Ilustración 5: Data Mart Financiero.</i> .....	24
<i>Ilustración 6: Data Mart Físico</i> .....	25
<i>Ilustración 7: Diseño Físico de la Arquitectura técnica.</i> .....	29
<i>Ilustración 8: Diseño lógico de la Arquitectura técnica.</i> .....	30
<i>Ilustración 9: Usuarios Finales.</i> .....	33
<i>Ilustración 10: Niveles de Organización en el SNIP.</i> .....	34
<i>Ilustración 11: Creación del DataSet.</i> .....	35
<i>Ilustración 12: DataSet creado.</i> .....	36
<i>Ilustración 13: Controles Radarsoft en el toolbox.</i> .....	36
<i>Ilustración 14: Componente RadarCube</i> .....	36
<i>Ilustración 15: estructura del Cubo.</i> .....	37
<i>Ilustración 16: TOLAPGrid enlazado con el RadarCube.</i> .....	37
<i>Ilustración 17:Charts de tacómetro configuradas con los indicadores de avance del SNIP</i> .....	39
<i>Ilustración 18: Herramienta de BI publicada en el portal del SNIP</i> .....	40
<i>Ilustración 19: Inicio de instalación de RadarSoft.</i> .....	58
<i>Ilustración 20: Radarsoft Licencia</i> .....	58
<i>Ilustración 21: Radarsoft Instalación</i> .....	59
<i>Ilustración 22: Habilitar controles de Radarsoft</i> .....	59
<input type="checkbox"/> <i>Ilustración 23: Creación del DataSet</i> .....	61

<i>Ilustración 24: Conexión de Datos.....</i>	<i>61</i>
<i>Ilustración 25: DataSet.....</i>	<i>62</i>
<i>Ilustración 26: Habilitar controles de Radarsoft.....</i>	<i>62</i>
<i>Ilustración 27: Controles de Radarsoft .....</i>	<i>62</i>
<i>Ilustración 28: Componente RadarCube .....</i>	<i>63</i>
<i>Ilustración 29: DataSet como fuente de datos.....</i>	<i>63</i>
<i>Ilustración 30: Estructura del Cubo .....</i>	<i>64</i>
<i>Ilustración 31: Estructura del Cubo .....</i>	<i>64</i>
<i>Ilustración 32: Estructura del Cubo .....</i>	<i>65</i>
<i>Ilustración 33: Estructura del Cubo .....</i>	<i>65</i>
<i>Ilustración 34: Estructura del Cubo .....</i>	<i>66</i>
<i>Ilustración 35: RadarCube con TolapGrid .....</i>	<i>67</i>
<i>Ilustración 36: RadarCube Desactivado.....</i>	<i>68</i>
<i>Ilustración 37: HightChart Gauge .....</i>	<i>82</i>
<i>Ilustración 38: Hightchart Columns.....</i>	<i>84</i>

# 1 INTRODUCCIÓN

El análisis de datos en las empresas es una necesidad cotidiana para guiar la toma de decisiones<sup>1</sup>. Este proceso ha tenido significativos cambios en las últimas décadas gracias a la llegada de herramientas tecnológicas para el soporte a la toma de decisiones que han facilitado la obtención, cruce e inferencia de información, tanto consolidada como al detalle, para los niveles intermedios y estratégicos de la organización.

Tales herramientas constituyen un dominio de aplicación al que se denomina en su conjunto “Inteligencia de Negocios” o BI por sus siglas en inglés (Business Intelligence). Según (Mendoza, 2008), ésta se define como una “Arquitectura y colección de herramientas que buscan mejorar a las organizaciones, proporcionando vistas de aspectos de negocio para que tomen mejores y más relevantes decisiones en menos tiempo y con la mayor información posible.”

El presente proyecto consistió en la implementación de una herramienta informática de Inteligencia de Negocios para apoyar a la Dirección General de Inversión Pública del Ministerio de Hacienda y Crédito Público en el análisis de datos para toma de decisiones a partir del seguimiento y monitoreo de la ejecución física y financiera de los proyectos que conforman el Programa de Inversiones Públicas de la República de Nicaragua.

---

<sup>1</sup> Según lo demostrado por (Rosado & Rico, Abril 2010) y (Morales, 2012).



## **2 OBJETIVOS**

### **2.1 Objetivo General**

Desarrollar una herramienta informática basada en técnicas de Inteligencia de Negocios, que integrada al sistema de información del Sistema Nacional de Inversión Pública (SNIP) de Nicaragua, facilite el análisis de información para monitoreo, seguimiento y evaluación de la ejecución de la inversión pública nacional.

### **2.2 Objetivos Específicos**

1. Familiarizarse con el cuerpo de conocimiento y técnicas de Inteligencia de Negocios disponibles que puedan emplearse para desarrollar una herramienta informática que facilite el análisis de los datos para monitoreo, seguimiento y evaluación de la ejecución de la inversión pública nacional.
2. Diseñar una herramienta informática que aproveche las técnicas de Inteligencia de Negocios para satisfacer las necesidades y requerimientos de los usuarios del Sistema Nacional de Inversión Pública, enfatizando en las capacidades de presentación ágil e interactiva de información consolidada empleando tanto tablas y matrices analíticas, como tableros de indicadores y gráficos.
3. Construir la herramienta informática sobre la base del diseño elaborado, aplicando los métodos y técnicas de la ingeniería de software, realizando las pruebas y aseguramiento de la calidad requeridos.
4. Implantar la herramienta en el sistema de información del Sistema Nacional de Inversión Pública y evaluar los resultados para verificar el cumplimiento de objetivos y beneficios esperados.

### 3 JUSTIFICACIÓN

Las herramientas de consumo de información actuales del Banco de Proyectos se limitan a: i) consultas consolidadas, ii) reportes fijos en archivos de Excel publicados en el portal del SNIP, y iii) reportes tabulares con agregación por grupos, con salida a Excel que se generan en el módulo de reportes. Este último, aunque es bastante parametrizable y versátil, es un tanto complicado de usar, y por su naturaleza estructural, a veces es requerido generar diferentes salidas y luego integrarlas para obtener la información requerida para ciertos análisis<sup>2</sup>.

Adicionalmente, todos estos instrumentos de aprovechamiento de información están orientados a la presentación de datos alfanuméricos, dificultando al nivel estratégico o de alta dirección, obtener una visualización significativa de los datos<sup>3</sup>. La alta dirección requiere además, interactividad en la consulta de los datos: partir del nivel más consolidado y poder ir profundizando o detallando la información sólo en las áreas problemáticas o de interés, para focalizar el esfuerzo del análisis.

Si se tiene en cuenta la importancia de monitorear y dar seguimiento continuo a la ejecución del Programa de Inversiones Públicas, generando información oportuna y alertas tempranas de los problemas y brechas a los tomadores de decisión, que permitan mejorar la eficiencia del proceso inversionista, y por ende de la gestión pública, entonces se hace evidente la necesidad de dotar al Banco de Proyectos del SNIP de una herramienta de inteligencia de negocios que venga a mejorar las capacidades actuales de aprovechamiento de información del sistema, facilitando y agilizando la obtención y análisis de información a los funcionarios del nivel estratégico.

---

<sup>2</sup> Según lo expresado por el Director de Desarrollo del Banco de Proyectos, responsable de la conceptualización, diseño e implementación de este sistema de información, en entrevista realizada por los autores.

<sup>33</sup> Según lo descrito por varios autores: (Andersen, 2013), (The tingley advantage, 2013) y (Tiwarei, 2014), los datos alfanuméricos obligan al usuario a realizar un proceso de análisis comparativo en base a cantidades, lo cual requiere mayor tiempo para generar significado, a diferencia de otras formas de presentación de datos como los gráficos o tableros de indicadores (Dashboards), que de un solo vistazo comunican mayor significado al usuario más rápidamente.

Al proporcionar al Banco de Proyectos del SNIP una herramienta de Inteligencia de Negocios, se espera obtener los siguientes beneficios para la Dirección General de Inversión Pública:

- Los funcionarios del nivel táctico no tendrán necesidad de generar múltiples reportes tabulares para obtener diferentes vistas de información.
- Se habrá facilitado el procedimiento de selección, consulta y obtención de información.
- Capacidad de visualizar datos de forma rápida, atractiva y eficaz a través de tableros de indicadores (Dashboards) y gráficos.
- Capacidad de analizar información dinámica e interactivamente.
- Mejorado el desempeño de la gestión al reducirse el tiempo de respuesta del sistema para la obtención de información<sup>4</sup>.

---

<sup>4</sup> Tomando como medida del desempeño el rendimiento, según lo establecido en (Henessy & Patterson, 1993), mismo que se define como el recíproco del tiempo de respuesta del sistema:  $R = 1/t$ .

## 4 MARCO TEÓRICO

Para abordar de forma adecuada el proceso de desarrollo de la herramienta de inteligencia de negocios propuesta, fue requerido revisar los conceptos relevantes del dominio de este campo, que permitiesen establecer un marco de referencia conceptual para el presente trabajo, mismos que se presentan a continuación:

### 4.1 Inteligencia de Negocios

A partir de lo planteado por (Mendoza, 2008), (Ayala, 2006), (Araníbar, 2003), (Caralt, Curto, & Conesa, 2010), se puede definir Inteligencia de Negocios o “*Bussiness Intelligence*” como una arquitectura y un conjunto de herramientas, metodologías, procesos y técnicas que contribuyen a la obtención y el análisis de la información para mejorar los procesos de toma de decisiones en una organización en menos tiempo y con la mayor información posible.

Los principales productos de Inteligencia de Negocios (BI por sus siglas en inglés) que existen hoy en día son<sup>5</sup>:

- **Cuadros de Mando Integral (CMI):** También conocido como Balanced Scorecard (BSC) o dashboard, es una herramienta de control empresarial que permite establecer y monitorizar los objetivos de una empresa y de sus diferentes áreas o unidades a partir de indicadores.
- **Sistemas de Soporte a la Decisión (DSS):** Herramienta enfocada al análisis de los datos de una organización para apoyar la toma de decisiones (comúnmente llamado Analítico).
- **Sistemas de Información Ejecutiva (EIS):** Es una herramienta basada en un DSS, que provee a los gerentes de un acceso sencillo a información que es relevante para sus factores clave de éxito.

---

<sup>5</sup> Tomado de (Sinergia e Inteligencia de Negocio S.L., 2015)

Por otro lado, los principales componentes de orígenes de datos para sistemas de inteligencia de negocios que existen en la actualidad son:

- **Datamart:** Es una base de datos especializada en una temática o área de negocio específica. Se puede poblar desde sistemas operacionales o desde un DataWareHouse.
- **DataWareHouse:** Es una base de datos corporativa que integra información depurada de una o más fuentes, que facilita su procesamiento para realizar análisis desde diversas perspectivas con grandes velocidades de respuesta.

Los sistemas de BI se orientan principalmente a apoyar a los niveles superiores de toma de decisiones de la Organización (estratégico y táctico), en tanto los sistemas operacionales (sistemas de gestión de información convencionales) se orientan al nivel operativo.

Los sistemas de BI se diferencian de los sistemas operacionales en que están optimizados para consultar y analizar datos. Esto implica que sus almacenes de datos están desnormalizados para apoyar consultas de alto rendimiento, mientras que en los sistemas operacionales suelen encontrarse normalizados para apoyar operaciones continuas de inserción, modificación y borrado de datos.

## 4.2 Niveles de toma de decisión en una organización

Establece (Anthony, 1965) que existen tres diferentes niveles de toma de decisiones dentro de una organización. Estas son la estratégica, la táctica y la operacional.

**Tabla 1: Niveles de Toma de Decisiones**

	Nivel Estratégico	Nivel Táctico	Nivel Operativo
<b>Tipos de decisiones</b>	Las metas de toda la organización, su	Desarrollo de tácticas que	Estas decisiones determinan la

	propósito y dirección	ayudan a que se alcancen las metas estratégicas definidas por la alta dirección.	manera en la cual las operaciones diarias son conducidas.
<b>Nivel de Gerencia</b>	Alta dirección	Niveles medios de la empresa, Ejemplo: jefes de departamento.	Nivel más bajo de la organización, ejemplo: supervisores o coordinadores.

### 4.3 Data Warehouse o Almacén de Datos

Según lo definido en (Oracle, 2014), un Data Warehouse es una base de datos relacional que contiene datos históricos y puede incluir datos de otras fuentes que está diseñado para la consulta y análisis. Separa el análisis de la carga de trabajo de transacciones y permite a una organización para consolidar los datos de varias fuentes.

Por su objetivo o aplicación podemos identificar dos tipos de Data Warehouse:

1. Data Warehouse Histórico: Es una base de datos que contiene todos los registros que se almacenan diariamente en la institución, es decir es un proceso enfocado al negocio.
2. Data Warehouse analítico: Son DataMart de los procesos más importantes de la organización.

#### 4.3.1 DataMart

Un DataMart (desde la perspectiva de Kimball) puede definirse como un componente de un DataWarehouse, ya que los DataMart son subsistemas que heredan todas las características de sus padres y a su vez pueden conformar un

sistema mismo (Oracle, 2014). Los DataMarts son a menudo contruidos y controlados por un solo departamento dentro de una organización. Dado su enfoque de un solo tema, los DataMarts normalmente consultan los datos de sólo unas pocas fuentes. Las fuentes pueden ser los sistemas internos de funcionamiento o datos externos.

#### **4.3.2 Modelado dimensional del Data Warehouse**

El modelo dimensional busca presentar la información de una manera estándar, sencilla y sobre todo intuitiva para los usuarios, además de que permite accesos a la información mucho más rápida por parte de los manejadores de base de datos.

Cada modelo dimensional está compuesto por una tabla llamada hechos y por un conjunto de pequeñas tablas llamadas dimensiones.

##### **4.3.2.1 Tablas de Hechos (Fact table):**

Las tablas de hechos se caracterizan por el nivel de detalle de la información que almacenan, contienen llaves foráneas a las tablas de dimensiones. Estas llaves foráneas relacionan cada fila de datos de la tabla de hechos con sus correspondientes dimensiones y niveles. La tabla de hechos contiene los datos numéricos de la aplicación.

##### **Tipos de Tablas de Hechos:<sup>6</sup>**

- **Tablas de Hechos de Transacciones** (Transaction Fact Tables): representan eventos que suceden en un determinado espacio-tiempo. Se caracterizan por permitir analizar los datos con el máximo detalle.
- **Tablas de hechos menores/Tablas de cobertura** (Factless Fact Tables/Coverage Tables): Son tablas que no contienen medidas,

---

<sup>6</sup>Tomado de: <http://data-warehouse.wikispaces.com/Tabla+de+Hechos>

representan el hecho que el evento suceda. Frecuentemente se añaden contadores a dichas tablas para facilitar las consultas SQL.

- **Tabla de Hechos instantánea periódica** (Periodic Snapshot Fact Tables): Son tablas de hecho usadas para recoger información de forma periódica a intervalos de tiempos regulares. Dependiendo de la situación medida o de la necesidad de negocio, ese tipo de tablas de hechos son una agregación de las anteriores o están diseñadas específicamente.
- **Tablas de hechos instantáneas de acumulación** (Accumulating Snapshot Fact Table): Representan el ciclo de vida completo de una actividad o proceso, que tiene un principio y final. Se caracterizan por presentar múltiples dimensiones que relacionadas con los eventos presentes es un proceso.

#### 4.3.2.2 Tabla de Dimensiones (Dimension table):

Una tabla de dimensiones almacena información descriptiva sobre los valores numéricos de una tabla de hechos. Corresponde a los clasificadores o categorizadores de la naturaleza de la información. Desde un punto de vista práctico se puede decir que corresponde a los catálogos de las diferentes variables que intervienen en la información.

#### 4.3.3 Tipos de Modelo Dimensional

Existen dos modelos dimensionales que predominan en las soluciones de Data Warehouse: Modelo en Estrella y Modelo en Copo de Nieve.

#### 4.3.4 Modelo en Estrella

Este modelo se caracteriza por contener una sola tabla de hechos central que está directamente relacionada a tablas de dimensiones.



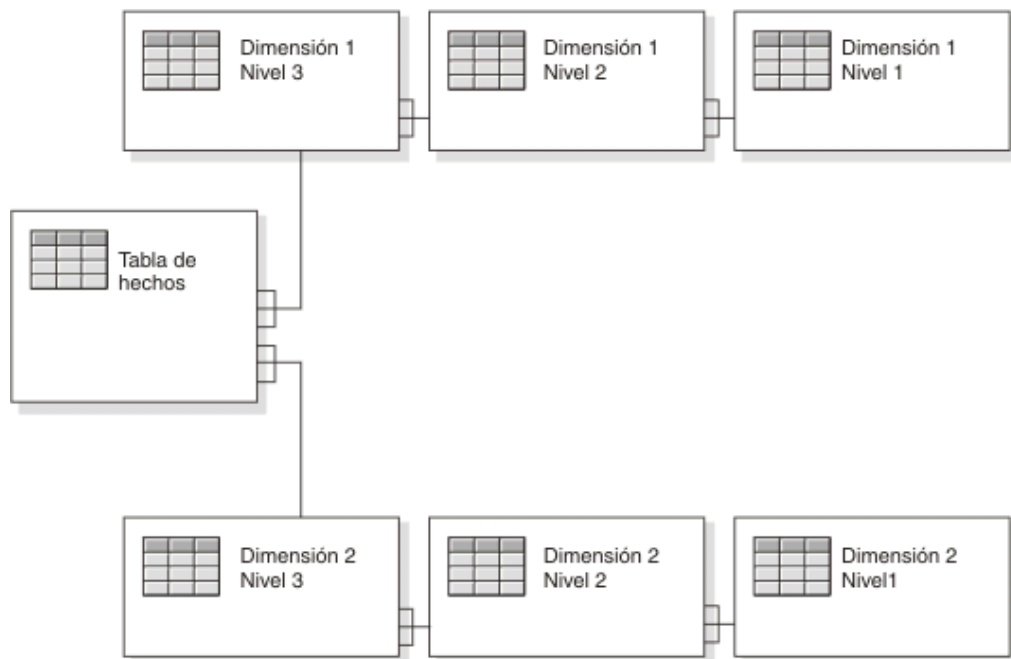
Ilustración 1: Esquema en Estrella



#### 4.3.5 Modelo en Copo de Nieve

En este modelo la tabla de hechos deja de ser la única relacionada con otras tablas ya que existen otras tablas que se relacionan con las dimensiones, esto hace que se vinculen más tablas de dimensiones haciendo la extracción de datos más difícil.

Ilustración 2: Esquema en Copo de Nieve



#### **4.3.6 ETL (Extracción, Transformación y Carga)**

ETL<sup>7</sup> es el proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos, Data Mart o Bodega de datos. ETL forma parte de la inteligencia empresarial (Business Intelligence) También llamado gestión de datos (Data Management).

### **4.4 Tipos de procesamiento de datos**

Existe básicamente dos tipos de procesamiento de datos: el transaccional (OLTP), que corresponde a la operación regular de los sistemas de gestión de información tradicionales orientados a la gestión del negocio, y el analítico (OLAP) que corresponde a las operaciones de análisis de grandes volúmenes de información generada por el procesamiento transaccional, orientada a caracterizar la dinámica de los datos para apoyar la toma de decisiones.

#### **4.4.1 OLTP: On Line Transactional Processing**

La (Academia latinoamericana de business intelligence [ALBI], 2010) define los sistemas de Procesamiento de transacciones en línea u OLTP por sus siglas en inglés como sistemas operacionales que guardan las transacciones realizadas en un negocio en Base de Datos.

Las características principales de los sistemas OLTP son:

- Realizan transacciones en tiempo real del proceso de un negocio, con lo cual los datos almacenados cambian continuamente.
- Los sistemas OLTP son los responsables del mantenimiento de los datos.
- Las estructuras de datos deben estar optimizadas para validar la entrada de los mismos.

---

<sup>7</sup> Tomado de: [etl-tools.info/es/bi/proceso](http://etl-tools.info/es/bi/proceso)

- Para la toma de decisiones, proporciona capacidades limitadas ya que no es su objetivo, por lo tanto no es prioridad en su diseño.

Se diferencian de los sistemas OLAP en cuanto a que aquellos tienen almacenados datos históricos para su consulta, mientras que éstos tienen los datos vivos que se están procesando por las transacciones del día a día, por lo que obtener información de éstos es más difícil y tardado de los sistemas OLAP.

#### **4.4.2 OLAP: On Line Analytical Processing**

Un sistema de Procesamiento analítico en línea u OLAP por sus siglas en inglés, según (Codd, Codd, & Salley, 1993), es un sistema de información para ejecutivos, que son utilizadas en el nivel estratégico para la toma de decisiones. Su idea principal es proporcionar navegación a través de los datos a los usuarios no expertos, a fin de que son capaces de generar interactivamente consultas ad-hoc sin la intervención de los profesionales de TI. En un modelo de datos OLAP, la información es vista como cubos, los cuales consisten de categorías descriptivas (dimensiones) y valores cuantitativos (medidas). El modelo de datos multidimensional simplifica a los usuarios formular consultas complejas, arreglar datos en un reporte, cambiar de datos resumidos a datos detallados y filtrar o rebanar los datos en subconjuntos significativos.

### **4.5 Dashboard (Panel de Indicadores Gerenciales)**

Un Dashboard de Inteligencia de Negocios o Cuadro de Mando Integral según (Rouse, 2010), es una herramienta de visualización de datos que muestra el estado actual de métricas e indicadores clave de rendimiento para una empresa en una sola pantalla. Pueden ser adaptados por roles un solo punto de vista o un departamento. Las características esenciales de un Dashboard de Inteligencia de Negocios son su interfaz personalizable y la capacidad de mostrar datos en tiempo real de múltiples fuentes.

## 4.6 Metodologías de desarrollo de Sistemas de Inteligencia de Negocios

Actualmente existen diversas metodologías que han surgido, de la mano de diversos autores, entre las que se destacan cuatro principales que se listan en la tabla 2, aunque predominan las de Inmon y de Kimball:

**Tabla 2: Metodologías de Diseño de DataWareHouse <sup>8</sup>**

	Top-Down	Bottom-Up	Hybrid	Federated
<b>Creador</b>	Inmon	Kimball	Muchos Profesionales	Doug Hackney
<b>Énfasis</b>	Data Warehouse	Data Marts	Data Warehouse y Data Marts	Integrado a entornos BI heterogéneos
<b>Diseño</b>	Modelo normalizado basado en la empresa.	El modelo dimensional de Data Marts, usa esquema estrella.	Modelos locales y uno o más esquemas de estrella.	Una arquitectura de arquitecturas: comparte dimensiones, hechos, reglas, definiciones a través de la organización.
<b>Arquitectura</b>	Compuesto de varios niveles de áreas de interés y	Área de interés y Data Marts	Modelo empresarial normalizado de alto nivel: Data Marts Iniciales	Realidad de cambio en organizaciones y sistemas

<sup>8</sup> Tomado de (IDORU, 2012)

	Data Marts dependientes			
<b>Data Set</b>	DWH: datos a nivel atómico.  DataMarts: datos sumariosados	Contiene datos atómicos y sumariosados	Carga Data Marts con datos atómicos y sumariosados vía un área de interés no persistente	Uso de cualquier significado posible para integrar las necesidades de negocio

Un aspecto importante que diferencia estas dos corrientes, desde el punto de vista arquitectónico, es la conceptualización del almacén o fuente de datos, punto de partida del sistema, y su sentido de construcción (Rivadera, 2010). Para Kimball, se parte del Datamart hacia el Data WareHouse (lo que se conoce como enfoque ascendente o Bottom-up). Por su parte, Inmon, parte del DataWarehouse, desde el principio, lo que se conoce como enfoque descendente o TopDown. Por tanto, desde esta óptica se tienen dos diferentes conceptos por parte de cada autor:

- **Bill Inmon:** "El Data Warehouse es una colección de datos orientados al tema, integrados, no volátiles e historizados, organizados para el apoyo de un proceso de ayuda a la decisión"<sup>9</sup>
- **Ralph Kimball:** "El Data Warehouse es una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis. En 1997 declaro que: un almacén de datos es la unión de todos los Data Marts".<sup>10</sup>

Se decidió emplear el enfoque de Kimball para el diseño e implementación del sistema de Inteligencia de Negocios propuesta. Esta metodología se basa en lo

<sup>9</sup> Texto de cita tomados de la referencia: (Inmon, 2002)

<sup>10</sup> Texto de cita tomados de la referencia: (Kimball & Ross, 2002)

que Kimball denomina Ciclo de vida dimensional del negocio. Este ciclo de vida del proyecto de DW está basado en cuatro principios básicos<sup>11</sup>:

- **Centrarse en el negocio:** Hay que concentrarse en la identificación de los requerimientos del negocio y su valor asociado, agudizando el análisis del mismo y la competencia consultiva de los implementadores.
- **Construir una infraestructura de información adecuada:** Diseñar una base de información única, integrada, fácil de usar, de alto rendimiento donde se reflejará la amplia gama de requerimientos de negocio identificados en la empresa.
- **Realizar entregas en incrementos significativos:** Priorizar los datos que confiere al negocio el mayor valor primero y luego ir progresivamente ampliando el orden de aplicación de los incrementos. En esto la metodología se parece a las metodologías ágiles de construcción de software.
- **Ofrecer la solución completa:** proporcionar todos los elementos necesarios para entregar valor a los usuarios del negocio. Requiere partir de un almacén de datos sólido, bien diseñado, con calidad y accesible; y sobre éste entregar herramientas de consulta ad hoc y aplicaciones para la eficaz explotación de la información.

La construcción de una solución de inteligencia de negocios puede llegar a ser sumamente compleja, por lo que esta metodología propone un proceso estructurado para simplificar esa complejidad. Este proceso parte de una cuidadosa planificación del proyecto y de una adecuada definición de requerimientos. Los requerimientos del negocio son el soporte inicial de las tareas subsiguientes. También tiene influencia en el plan de proyecto.

Luego de esta etapa inicial, el proceso se desagrega en tres rutas o caminos que se enfocan en tres diferentes áreas:

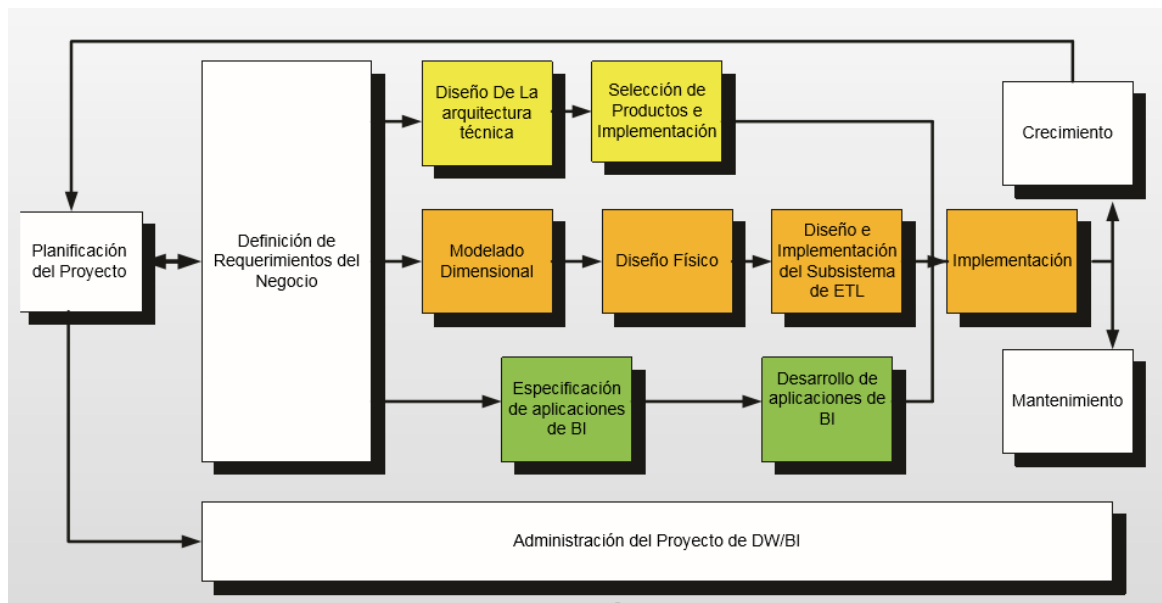
- **Tecnología** (Camino Superior). Implica tareas relacionadas con la arquitectura y selección de productos.

---

<sup>11</sup> Ver referencia: (Rivadera, 2010)

- **Datos** (Camino del medio). Corresponde al diseño e implementación el modelo dimensional, así como el desarrollo del subsistema de Extracción, Transformación y Carga (ETL).
- **Aplicaciones de Inteligencia de Negocios** (Camino Inferior). En esta ruta se encuentran tareas de diseño y desarrollo de las aplicaciones de negocios para los usuarios finales. Las tareas de este proceso se detallan a continuación:

Ilustración 3: Ciclo de vida del sistema BI, Metodología de Kimball.



## 4.7 Proceso de desarrollo de Software

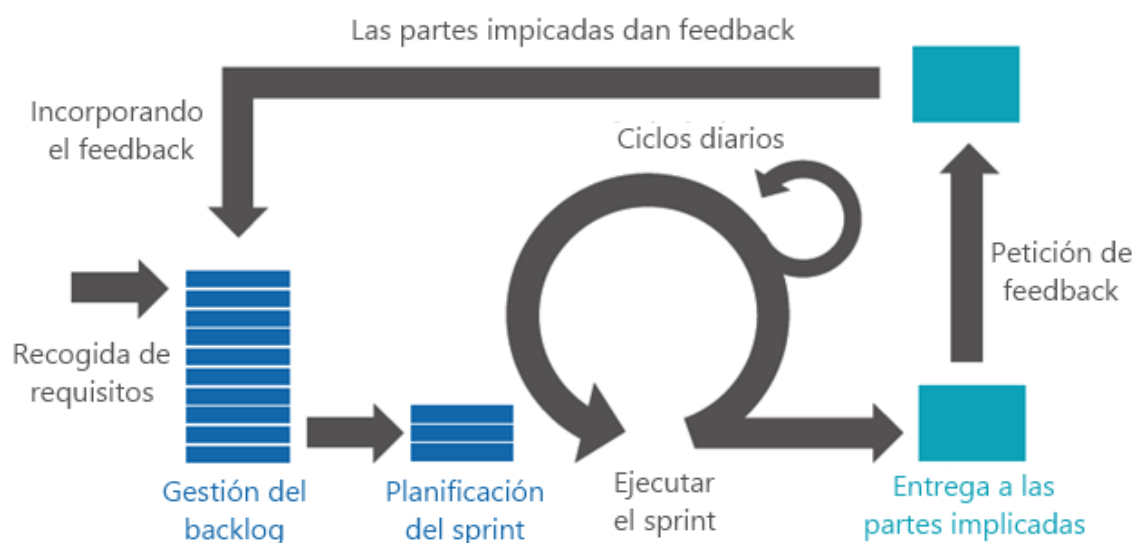
Para complementar la metodología de desarrollo BI, para el desarrollo de las aplicaciones de Inteligencia de Negocio propuestas se decidió complementar el proceso de desarrollo de software con la metodología ágil Scrum. Ésta se caracteriza por la rapidez y el orden con que se trabaja. Se ha empleado para gestionar el desarrollo de productos complejos de software. Scrum no es un proceso o una técnica para construir productos, más bien es un marco de trabajo dentro del cual se pueden emplear diversas técnicas y procesos (Shutherland & Shwaber, 2013)

El marco de trabajo Scrum consiste en los Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de Scrum y para su uso. Permite definir asignaciones y roles para los involucrados en todo el ciclo de vida del proyecto, así también tareas e interacciones a ejecutarse en tiempos específicos.

De esta metodología ágil, el flujo de trabajo y algunos instrumentos (como la tabla de roles y el backlog), son apropiados para complementar la metodología de Inteligencia de Negocios de Kimball en el proceso de desarrollo de sistemas de BI.

El flujo de trabajo que establece SCRUM<sup>12</sup> se describe en la ilustración que se muestra a continuación:

Ilustración 4: Flujo de Trabajo de la Metodología Scrum.



<sup>12</sup> SCRUM es una metodología ágil de gestión de proyectos de software. La información descrita en este documento fue obtenida de las siguientes fuentes: (Shutherland & Shwaber, 2013), (Escolar, 2013) y (proyectosagiles.org, 2015)



## 5 ANALISIS Y PRESENTACIÓN DE RESULTADOS

### 5.1 Metodología Empleada

Una parte sumamente importante para el desarrollo de la solución de inteligencia de negocios, es el almacén de datos o Data Warehouse, que en este tipo de aplicaciones conforma el corazón del sistema. En este tipo de aplicaciones, a diferencia de las soluciones software tradicionales, se requiere para su planificación e implementación una metodología propia de Inteligencia de Negocios. Sobre la base de lo anterior, para el desarrollo del presente proyecto se decidió emplear una metodología híbrida que se basa en la integración del proceso de organización del trabajo que provee la metodología SCRUM con la metodología de implementación de Data Warehouse que propone Kimball.

### 5.2 Definición de roles del proyecto según SCRUM

La implantación de la metodología ágil SCRUM requiere de definir apropiadamente las personas involucradas con sus respectivos roles en el proyecto los cuales se detallan en la siguiente tabla.

**Tabla 3: Personas y roles del proyecto**

Persona	Contacto	Rol
<b>Judith Navarro,</b> <b>José Zamora</b>	judith-navarro@hotmail.es jlzo21@hotmail.com	Desarrolladores del Producto
<b>Ing. Jorge Ulises</b> <b>Arróliga</b> <b>(Director de</b> <b>desarrollo del Banco</b> <b>de Proyectos SNIP)</b>	jarroliga@snip.gob.ni	Cliente

## **5.3 Desarrollo de la Solución**

Según lo planteado anteriormente, el desarrollo e implantación de la solución se realizó siguiendo la metodología de Kimball, por lo cual se reporta en el presente documento de acuerdo al ciclo de vida de dicha metodología, según se detalla a continuación:

### **5.3.1 Planificación del proyecto**

Se realizaron reuniones con el Director de desarrollo del Banco de Proyectos SNIP para identificar actividades, recursos y tiempos en el desarrollo del proyecto.

### **5.3.2 Definición de los requerimientos del negocio**

En las sesiones de trabajo anteriormente mencionadas, se definieron los requerimientos del negocio y las herramientas para los niveles de gestión, estratégico y táctico.

#### **5.3.2.1 Requerimientos funcionales**

1. Se requiere que la solución de Inteligencia de Negocios se base sobre almacenes de datos para 2 procesos específicos:
  - Financiero
  - Indicadores físicos
2. Se requiere que la herramienta integre funcionalidades de sistema de soporte a decisión (SSD) o analítico, sobre la base de una tabla dinámica o pivote interactiva donde el usuario pueda seleccionar un almacén específico de datos y parametrizar consultas sobre el almacén, basadas inicialmente en una selección de años, y posteriormente poder, de forma interactiva, seleccionar las dimensiones deseadas para generar vistas diferentes de la información, a diferentes niveles de agregación, sobre la misma tabla dinámicamente.

3. Se requiere asimismo, que la herramienta contenga funcionalidades de un cuadro de mando integral (CMI) que integre tableros dinámicos que permitan mostrar el avance de los indicadores de la ejecución del Plan de Inversiones Públicas que apoye a los procesos de monitoreo y seguimiento de esta ejecución.

#### 5.3.2.2 Requerimientos no funcionales

1. **Requerimientos de tipo de Tecnología:** Se requiere que la herramienta de Inteligencia de Negocios sea una aplicación web a fin que pueda ser fácilmente desplegada, actualizada y accedida por los usuarios tanto a lo interno de la institución como desde el exterior.
2. **Multiusuario:** Se requiere que se pueda operar por múltiples usuarios de forma simultánea y concurrente.
3. **Facilidad de uso:** Se requiere que la interfaz gráfica sea lo más simple y atractiva posible, a fin de permitir al usuario comprender e interactuar de forma sencilla con el software.
4. **Seguridad:** Se requiere que la implementación de la seguridad deberá orientarse a nivel de formulario (pantallas de la aplicación) con el marco de trabajo de seguridad de usuarios que ya está implementada en la Dirección General de Inversión Pública, por lo cual, el alcance de la aplicación para este proyecto no incorpora restricciones de seguridad. Ésta será integrada posteriormente por la Dirección de Desarrollo del Banco de Proyectos, unidad informática a cargo del desarrollo del software del SNIP.
5. **Rendimiento:** Se requiere que se realice un modelo dimensional de tipo estrella, para la facilidad y rapidez de consultas.
6. **Fiabilidad:** Se requiere que la herramienta tenga mecanismos de recuperación frente a fallos del sistema.
7. **Soporte:** Se requiere que el software esté documentado técnicamente para facilitar su mantenimiento y actualización por parte de la Dirección de Desarrollo del Banco de Proyectos.

### 5.3.3 Definición del Backlog inicial de SCUM

El flujo de trabajo de SCRUM parte de la obtención o recogida de los requisitos del sistema, esto se realizó mediante reuniones de trabajo con el cliente o sus representantes, que en este caso fue el ingeniero Jorge Ulises Arróliga Téllez, Director de desarrollo del Banco de Proyectos, que tiene a cargo la gerencia del área de desarrollo de software para el SNIP.

A partir de los requerimientos del cliente y los procesos propuestos en la Metodología de Kimball, se definió la programación de tareas a realizar, lo que se recoge una lista denominada Backlog que es el instrumento de organización central del trabajo en la metodología SCRUM.

**Tabla 4: Tareas Programadas en el Backlog inicial**

Sprint <sup>13</sup>	Tareas Programadas
0	Planificación del proyecto
1	Definición de los requerimientos del sistema
2	Modelado dimensional
3	Diseño y desarrollo de la presentación de datos
4	Selección de productos e instalación
5	Diseño e implementación del almacén de datos.
6	Implementación de la herramienta de soporte a decisión (DSS o analítico).
7	Investigación sobre herramientas de software para la implementación de la herramienta de cuadro de mando integral (CMI o Dashboard).
7	Desarrollo del dashboard.

<sup>13</sup> Sprint es el corazón de Scrum. Corresponde a un ciclo completo del flujo de trabajo que se realiza en un bloque de tiempo (time-box) durante el cual se crea un incremento de producto terminado, utilizable y potencialmente entregable (Shutherland & Shwaber, 2013).

Sprint <sup>13</sup>	Tareas Programadas
8	Especificación de aplicaciones para usuarios finales.
9	Mantenimiento y crecimiento del Data Warehouse
10	Evaluación

### 5.3.4 Modelado dimensional

Se realizó un proceso iterativo de alto nivel para definir los procesos más importantes del negocio, las dimensiones de cada indicador, los diferentes grados de detalle (atributos), así como la granularidad de cada indicador y las jerarquías que dan forma al modelo dimensional del negocio.

#### 5.3.4.1 Elegir el proceso del Negocio

El banco de datos del SNIP está integrado por múltiples secciones de datos, las que reflejan la información de rigor de los diferentes procesos de inversión en proyectos públicos del Gobierno. Basado en lo anterior y en los requisitos del negocio se estableció con el director Desarrollo del Banco de Proyectos del SNIP, para el alcance de este proyecto, modelizar dos áreas importantes:

- Sección Financiera.
- Sección de Indicadores Físicos.

#### 5.3.4.2 Establecer el nivel de granularidad

La base de datos del Banco de Proyectos del SNIP tiene una arquitectura relacional y con modelo de datos mayormente recursiva en lo que respecta a los catálogos, posee datos redundantes debido a múltiples ejercicios y nulos, dificultando la extracción de la información ya que para llegar a un dato específico hay que ejecutar grandes consultas, aumentando el tiempo de respuesta en la ejecución de las mismas. Para ello se realizó un análisis sobre

la BD histórica del SNIP especificando el mayor nivel de detalle posible para la alta gerencia a través del modelo dimensional de tipo estrella. Este modelo incluye procesos tales como:

- Selección de los datos (Se definen las tablas de interés)
- Depuración de los datos (depuración de datos redundantes)
- Estructuración de los datos (Generación de atributos y registros)
- Integración de los datos (Agrupar los datos)

#### 5.3.4.3 Elegir las dimensiones

En este proceso se seleccionaron aquellas tablas que describen una medida en una tabla de hechos. En el caso de la información de la Inversión pública, la principal dimensión es el **Proyecto**, el cual se estructura en **Actividades** (como estudios, supervisión de obras, adquisiciones y la administración del proyecto) y **Obras** de implementación. Los proyectos son ejecutados por **Instituciones** del Estado, las cuales se pueden clasificar en **Sectores institucionales**. Asimismo, los proyectos contribuyen al capital de la nación de un **Sector Económico**, son financiados por diferentes **Fuentes** mediante **Convenios** en diferentes **Modalidades** de financiamiento (donaciones o préstamos), realizando ejecución en diferentes **Renglones** de gastos y ubicaciones geográficas (**Departamento, Municipio**). Asimismo, genera **Productos** (bienes o servicios) de diferentes **Tipologías de Inversión** (construcción de carreteras, de escuelas, rehabilitación de hospitales, etc.).

#### 5.3.4.4 Identificar las tablas de hechos y medidas.

Para este proceso se identificaron únicamente dos tablas de hechos: financiera y física, cada tabla de hechos tiene como atributo uno o más medidas del proceso organizacional del banco de datos del SNIP. La tabla de hechos financiera se decidió implementar en dos niveles de detalle de desagregación de tiempo: por año y por mes.

### 5.3.4.5 Modelo grafico de Data Marts

A continuación se muestra el modelado dimensional de los Data Marts Financiero y Físico.

Ilustración 5: Data Mart Financiero.

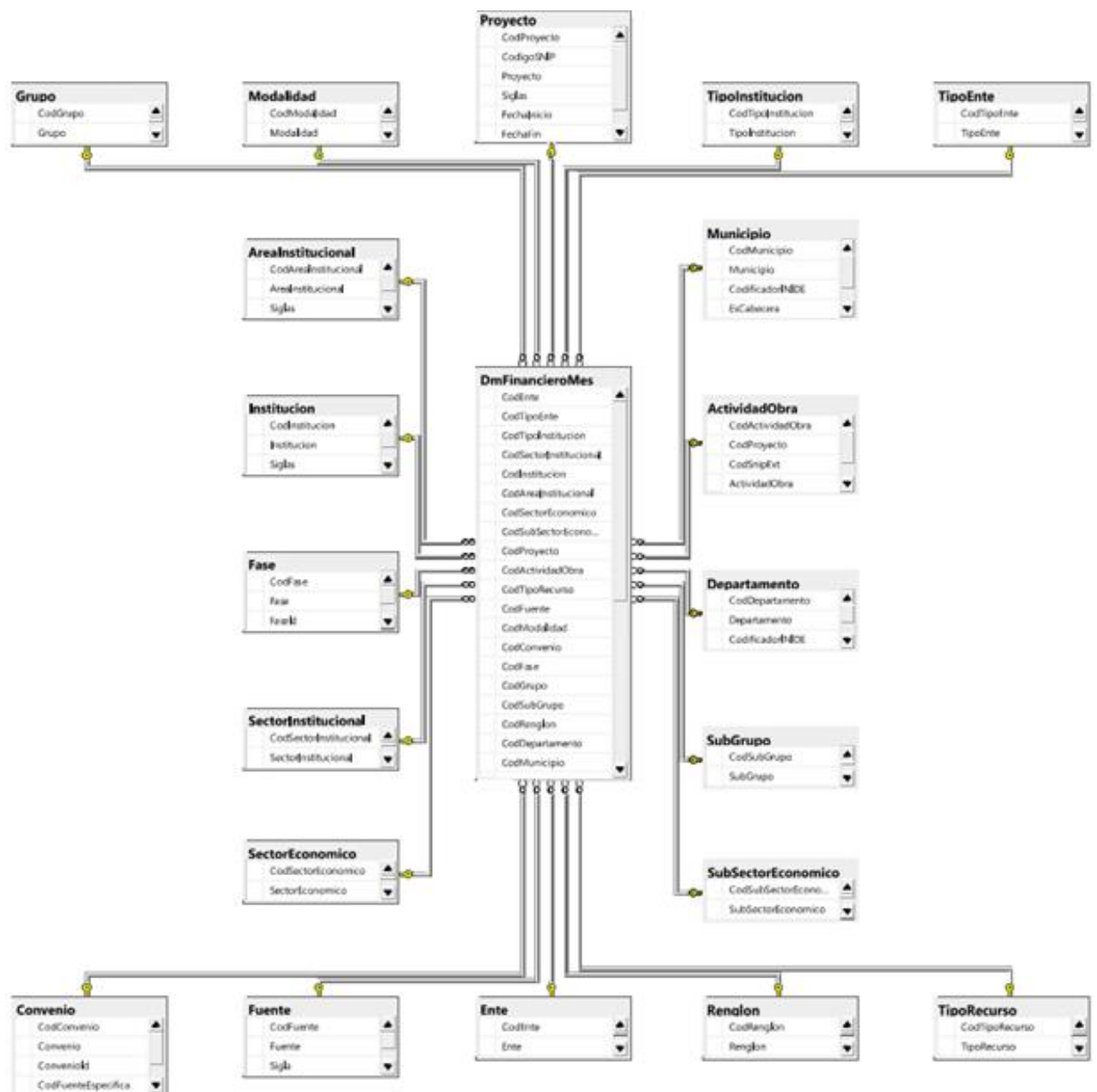
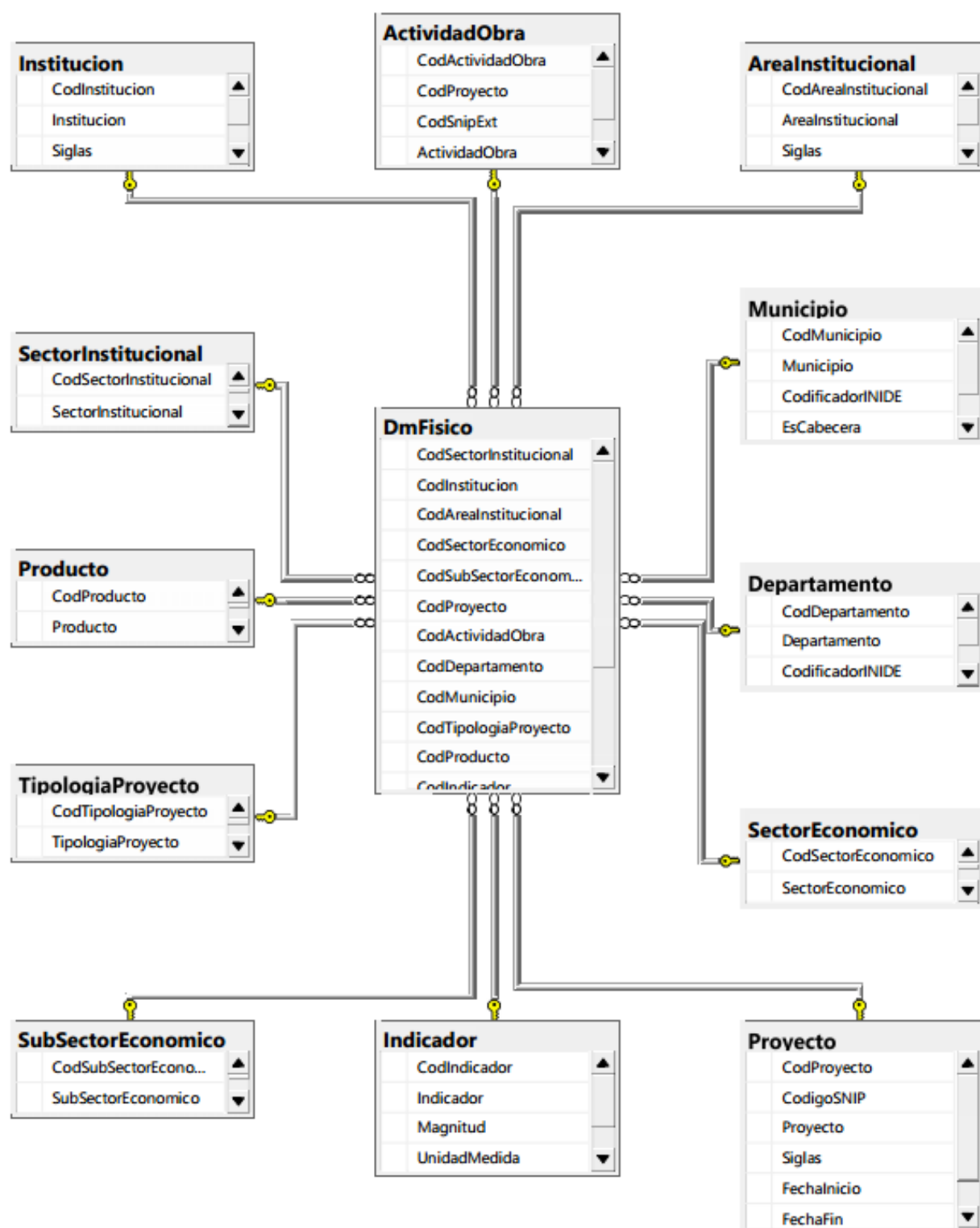


Ilustración 6: Data Mart Físico





### 5.3.5 Diseño físico

Kimball propone realizar el siguiente cuestionario para obtener los insumos necesarios para realizar un buen diseño Físico. Se listan a continuación las preguntas que proponen la metodología y las respuestas obtenidas del cliente:

#### 1. ¿Cuán grande será el sistema de DW/BI?

Tomando en cuenta que la base de datos histórica actual del SNIP cuenta con 2.7 GB pero mucha de su información corresponde a simulaciones y proyecciones, los datos oficiales que son los que se integrarán al almacén de datos no debería superar 1.5 GB.

#### 2. ¿Cuáles son los factores de uso que llevarán a una configuración más grande y más compleja?

Los patrones de crecimiento de los proyectos de inversión pública: Más proyectos, proyectos diferenciados en más actividades u obras, proyectos con mucha dispersión geográfica, proyectos con mucha información de gastos operativos. De cualquier forma, de acuerdo a la historia no debería crecer más de 200 MB por año.

#### 3. ¿Cómo se debe configurar el sistema?

De acuerdo a las limitaciones de recursos de DGIP, el almacén de datos se alojará en el mismo servidor de la base de datos transaccional del Banco de Proyectos pero empleando una base de datos separada denominada SnipDM.

#### 4. ¿Cuánta memoria y servidores se necesitan? ¿Qué tipo de almacenamiento y procesadores?

Se requiere un despliegue en producción en dos servidores: Un servidor de base de datos y un servidor de aplicaciones en el cual se alojarán

ambas herramientas de inteligencia de negocios (Analítico y Dashboard) consultando la información del almacén de datos. Adicionalmente los trabajos de ETL se harán en el servidor de base de datos mediante el uso de scripts disparados por trabajos programados extrayendo los datos de la base de datos transaccional, transformándolos y cargándolos en la base de datos del Almacén.

- El **Servidor de Base de datos** es un servidor de clase empresarial con Procesador Intel Xeon de 4 núcleos a 2.5GHz x 2 con HiperThreading activado para un total de 8 núcleos y 16 procesos simultáneos. Memoria principal operativa de 16 GB en RAM y almacenamiento redundante basado en 5 discos SAS de 15Krpm en RAID 5 con espacio de 688 GB para datos.
- El **Servidor de aplicaciones** con que se dispone es un Servidor web equipado con IIS 7.0 y tecnología del Framework .Net 4. Es un servidor de clase empresarial con Procesador Intel Xeon de 4 núcleos a 2.5GHz x 2 con HiperThreading activado para un total de 8 núcleos y 16 procesos simultáneos. Memoria principal operativa de 16 GB en RAM y almacenamiento redundante basado en 4 discos SAS de 15Krpm en RAID 5 con espacio de 516 GB para datos.

En desarrollo y prueba se emplean ambos roles en una misma computadora, en este caso, una computadora portátil equipada con un procesador AMD A8 de 4 núcleos a 1.7 GHz con 8 GB de memoria principal y 700 GB de disco duro.

##### **5. ¿Qué necesitan instalar los diferentes miembros del equipo de DW/BI en sus estaciones de trabajo?**

No se requiere instalar software alguno porque la herramientas de inteligencia de negocio que se están desarrollando son basadas en

web y por tanto solo es requerido tener acceso a la red y un navegador de internet.

### **5.3.6 Diseño y desarrollo de la presentación de datos**

La aplicación del ETL realiza tres pasos críticos que dependen uno del otro:

#### **1. Proceso de Extracción:**

En este proceso se tomó como fuente de datos de origen la BD del SNIP, donde se realizó una selección de tablas para crear una nueva estructura de la BD que facilite el acceso de los datos con mayor nivel de detalle.

#### **2. Proceso de Transformación:**

Los datos extraídos son transformados mediante la aplicación de las reglas de negocio y procedimiento de transformación, A continuación se presentan algunas transformaciones:

- Definir una etiqueta para las tablas
- Generar reportes de los proyectos que se ejecutan en la tabla financiera por cada mes del año.
- Dividir una columna en varias
- Obtener Nuevos valores calculados
- Población de las tablas de hechos y dimensiones

#### **3. Proceso de carga:**

En esta fase, los datos precedentes del proceso anterior (proceso de transformación) son cargados en el sistema destino Base de Datos del Data Warehouse, este sistema permite cargar los datos de las tablas como valores y atributos, luego son filtrados a través de una consulta para luego ser presentados en modo reporte por el Sistema de Soporte a Decisiones (SSD o Analítico).

### 5.3.7 Diseño de la arquitectura técnica

#### 5.3.7.1 Diseño físico

El diseño físico, en lo concerniente a la arquitectura técnica, muestra la arquitectura de despliegue del sistema de Inteligencia de Negocios en los servidores de la organización o su proveedor de servicios y cómo los acceden los usuarios finales. La Ilustración 7 muestra el diseño físico de la herramienta propuesta:

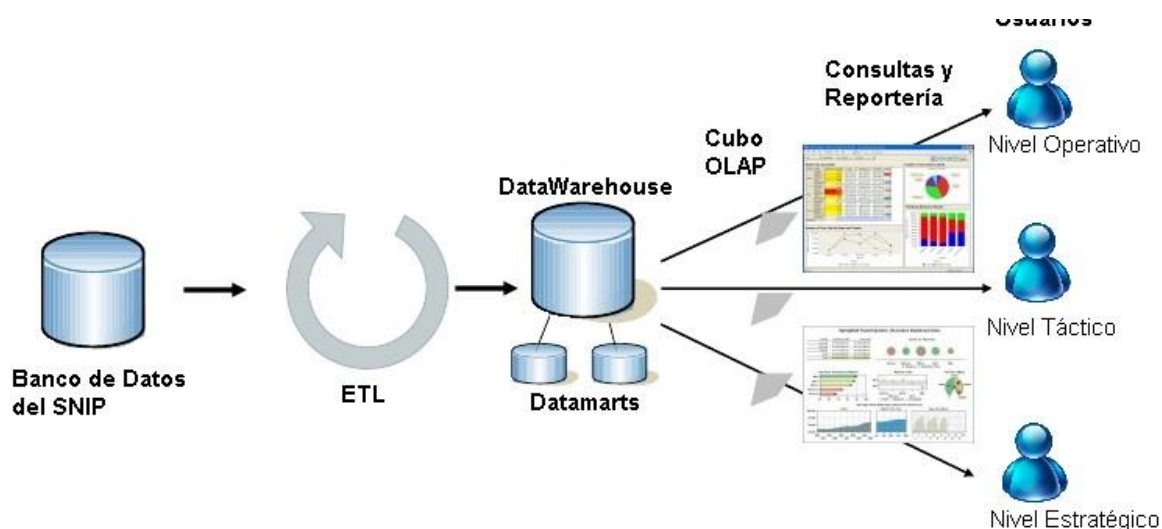
Ilustración 7: Diseño Físico de la Arquitectura técnica.



#### 5.3.7.2 Diseño Lógico

El diseño lógico, en cuanto a la arquitectura técnica se refiere, muestra el proceso de trabajo o procesamiento del sistema de Inteligencia de Negocios, partiendo de las fuentes de datos, el proceso de extracción, transformación y carga de datos, almacenado en el Data Warehouse y el proceso de análisis y presentación al usuario final. La Ilustración 8 muestra el diseño lógico de la arquitectura técnica de la herramienta propuesta:

Ilustración 8: Diseño lógico de la Arquitectura técnica.



### 5.3.8 Selección de productos e instalación

Para el desarrollo de la solución, la DDBP definió la restricción que la aplicación de BI debe integrarse al resto de aplicativos del SNIP, por lo cual su desarrollo debería de realizarse sobre la misma plataforma que estas: Microsoft.Net Framework 3.5 con desarrollo en Visual Studio 2010 empleando lenguaje C#, accediendo a Base de Datos SQL Server 2005. El resto de componentes de desarrollo para facilitar la implementación de la herramienta quedo a decisión del equipo de desarrollo. Así mismo la aplicación requiere ser web, desarrollada como aplicación ASP.Net con Web Forms.

Para el desarrollo del analítico se consideraron las siguientes suites de componentes:

- a. **Syncfusion**<sup>14</sup> es una plataforma que ofrece herramientas de software gratis para ASP.Net, a través de la licencia Community Edition, para equipos con hasta cinco desarrolladores, incluyendo el uso comercial. Contiene componentes para presentación de datos de inteligencia de negocios como:

<sup>14</sup> <https://www.syncfusion.com/>

- **PivotGrid.** Es una tabla dinámica que permite la visualización de datos relacionales, brinda la opción de exportar la información en formatos de Office (Excel, Word y documentos).
- **OLAP Client.** Incluye una interfaz de usuario de gran alcance para la gestión de datos OLAP con el análisis de inteligencia de negocios que incluye filtros, resúmenes y KPIs.

Estos controles tienen la ventaja de poder manejar grandes cantidades de datos estructurados, de forma dinámica y analítica, así como visualizar en formato gráfico información consolidada a través de indicadores. Tienen excelentes características y estilos visuales. Sin embargo requieren leer la información de una fuente OLAP de Microsoft SQL Server Análisis Services (MS SSAS), lo que es una desventaja para este proyecto, ya que la institución decidió no usar los servicios MS SSAS, puesto que demanda mayores recursos de hardware con los que no se cuentan en la actualidad.

- b. **Radarsoft:** Esta herramienta se propuso para el desarrollo del cubo analítico, la institución posee la licencia de uso, pero nunca se había utilizado, considerando que es compatible con las herramientas seleccionadas (Visual Studio C#) se aprovechará y se hará uso de la misma.

La herramienta Radarsoft permite realizar cubos dinámicos con Data Warehouse, utiliza filtros para visualizar información detallada, que a diferencia de syncfusion no requiere de un servidor de MS SSAS lo cual representa una ventaja al no saturar el servidor del SNIP.

Para la implementación del Dashboard o cuadro de mando integral, fueron estudiadas las siguientes herramientas:

- a. **Highcharts:** es una librería de JavaScript para la creación de gráficos, es libre para proyectos de uso no comercial. Entre los gráficos a utilizar están:
  - Gráficos de líneas

- Gauges
- Gráficos de Pasteles
- Gráficos de Barras

Entre las ventajas que presenta esta herramienta es su integración con el lenguaje c#, permite agregar gráficos interactivos a un sitio web, es flexible y fácil de usar, no requiere plugins secundarios como Flash o Java, no se necesita instalar ningún componente en el servidor. Basándose en lo mencionado anteriormente, cabe destacar que Highcharts fue la herramienta utilizada para la elaboración de tableros.

- b. **Syncfusion:** ofrece algunos componentes gratis para la elaboración de tableros, entre ellos:

- OLAP Chart y Gauge

Son capaz de representar los datos de inteligencia de negocios como gráficos, su principal desventaja es que solamente son compatibles con XMLA como MS SQL y Mondrian BI.

- c. **Dundas Charts:** Provee gráficos para OLAP y Charting. Se instalan para ser usados en Visual Studio. Entre sus desventajas, es una herramienta propietaria y su uso está restringido solo al propietario de la licencia. El SNIP cuenta con una licencia de Dundas pero es una versión antigua.

### 5.3.9 Especificación de aplicaciones para usuarios finales

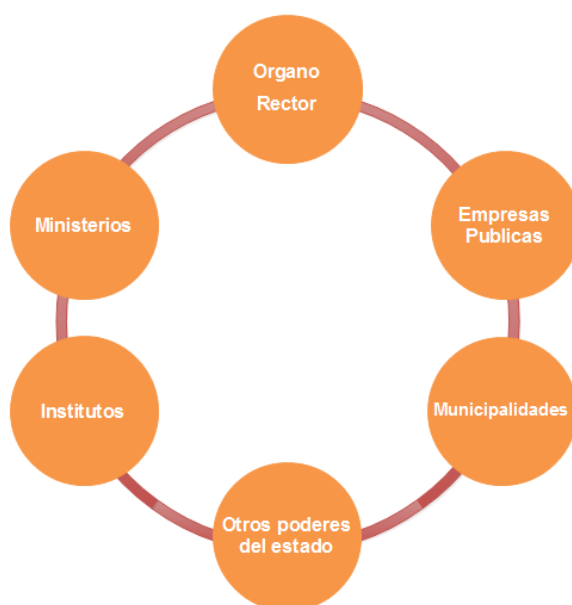
Los niveles de toma de decisión presentes en una organización estándar son: el estratégico o de alta dirección, el táctico o de gerencia y el operativo. Fue necesario seleccionar las herramientas de inteligencia de negocios idóneas para cada nivel a fin que cada uno de ellos sea afectado de manera positiva en la mejora de su gestión.

El nivel operativo es el que está en contacto con las transacciones diarias del negocio y por tanto sus requerimientos de información son siempre al más mínimo detalle por lo que requiere herramientas de reportes dinámicos.

El nivel táctico requiere analizar las operaciones a niveles agregados desde la perspectiva de múltiples variables o dimensiones de información por lo cual requiere herramientas analíticas como tablas dinámicas.

El nivel estratégico requiere monitorear los indicadores de gestión del negocio para lo cual requiere un nivel de mayor agregación de la información mediante herramientas graficas de visualización de datos (Tacómetros y gráficos de barra, pasteles y cuadros de mando).

Ilustración 9: Usuarios Finales.



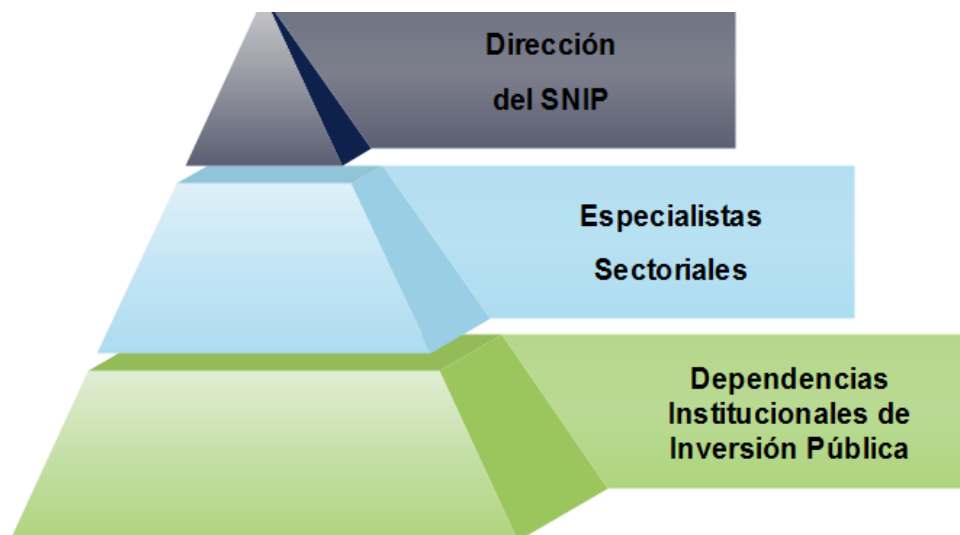
El SNIP es un sistema administrativo que se conforma por cada una de las instituciones que del estado que realiza inversión pública<sup>15</sup>, coordinados por su órgano rector que es la Dirección General de Inversión Pública (DGIP) del Ministerio de Hacienda y Crédito Público (MHCP).

<sup>15</sup> Se entiende por inversión pública toda acción que involucra recursos para ampliar o mantener el stock de capital del estado.



Los niveles estratégicos y tácticos de SNIP se concentran en el órgano rector y el nivel operativo en la institución ejecutora.

Ilustración 10: Niveles de Organización en el SNIP.



#### 5.3.10 Desarrollo de aplicaciones para usuarios finales

Para el nivel táctico se implementó una herramienta analítica del tipo Sistema de Soporte a Decisión (SSD) basada en web, conformada por tres cubos OLAP (Financiero anual, Financiero mensual y de Indicadores Físicos) y una tabla dinámica que el especialista sectorial y los directores puedan utilizar para ejecutar cruces de información. Para el nivel estratégico se creó un Cuadro de Mando Integral (CMI) que muestra el estado de los indicadores de la gestión de inversión pública.

El nivel operativo ya cuenta con las herramientas de generación de reportes del Banco de Proyectos.

A continuación se describe de forma breve el proceso de implementación de estas dos herramientas de Inteligencia de Negocios que conforman la propuesta presentada a DGIP.

### 5.3.10.1 Implementación de analítico

El cliente decidió agregar un módulo de inteligencia de negocios en la página web que actualmente utilizan, la cual será administrada por los usuarios de nivel estratégico. Este módulo muestra una plataforma accesible para el usuario facilitando las medidas y los atributos que el cubo requiere para mostrar información a través de reportes dinámicos.

La herramienta consiste en una aplicación web con un simple menú para seleccionar el Data Mart a consultar (cubo de datos) y luego un panel de selección o parametrización de la consulta por dimensiones. Esta aplicación se implementó sobre los componentes OLAP de RadarSoft en modo Direct (cubos en memoria).

El proceso de implementación siguió los siguientes pasos:

- Crear un nuevo proyecto de Aplicación web en Visual Studio.
- Crear un DataSet y seleccionar la BD del Data Warehouse como fuente.

Ilustración 11: Creación del DataSet.

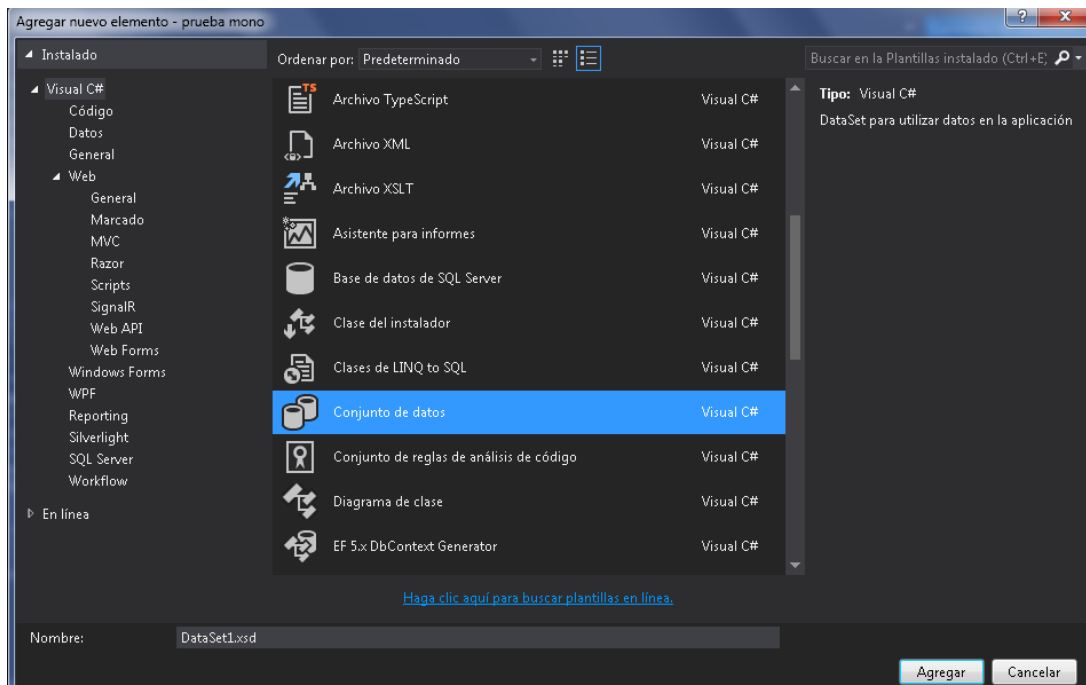
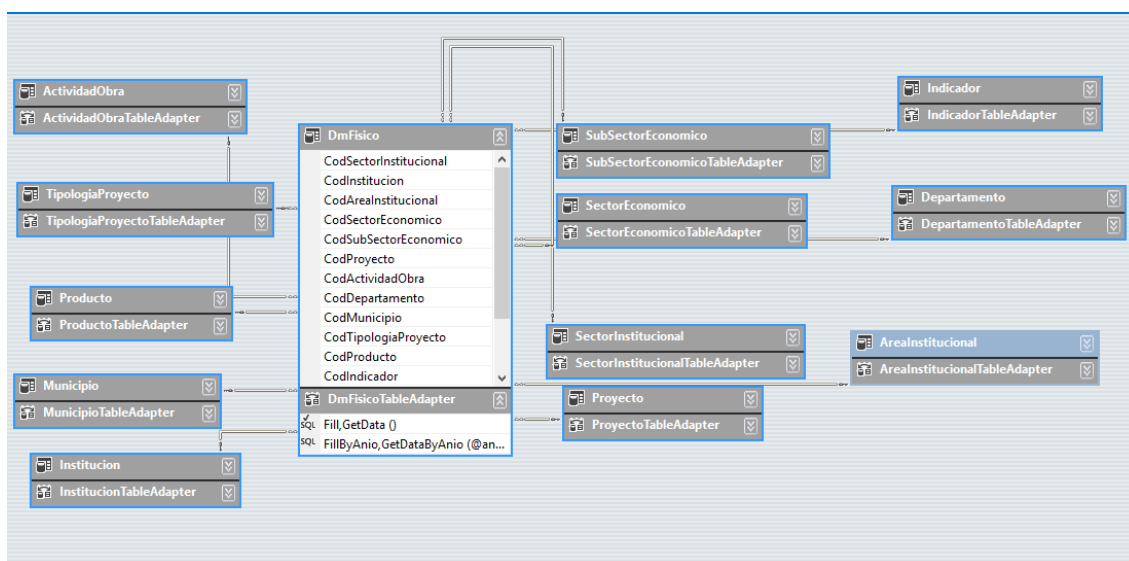
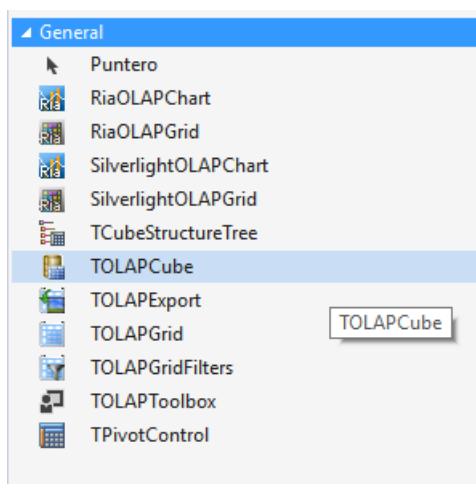


Ilustración 12: DataSet creado.



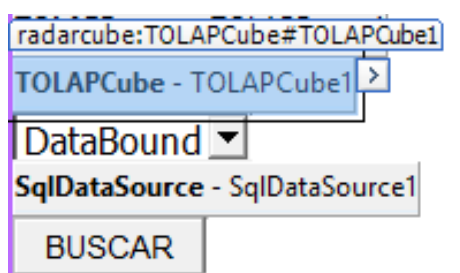
- Activar los controles de Radarsoft en el proyecto.

Ilustración 13: Controles Radarsoft en el toolbox.



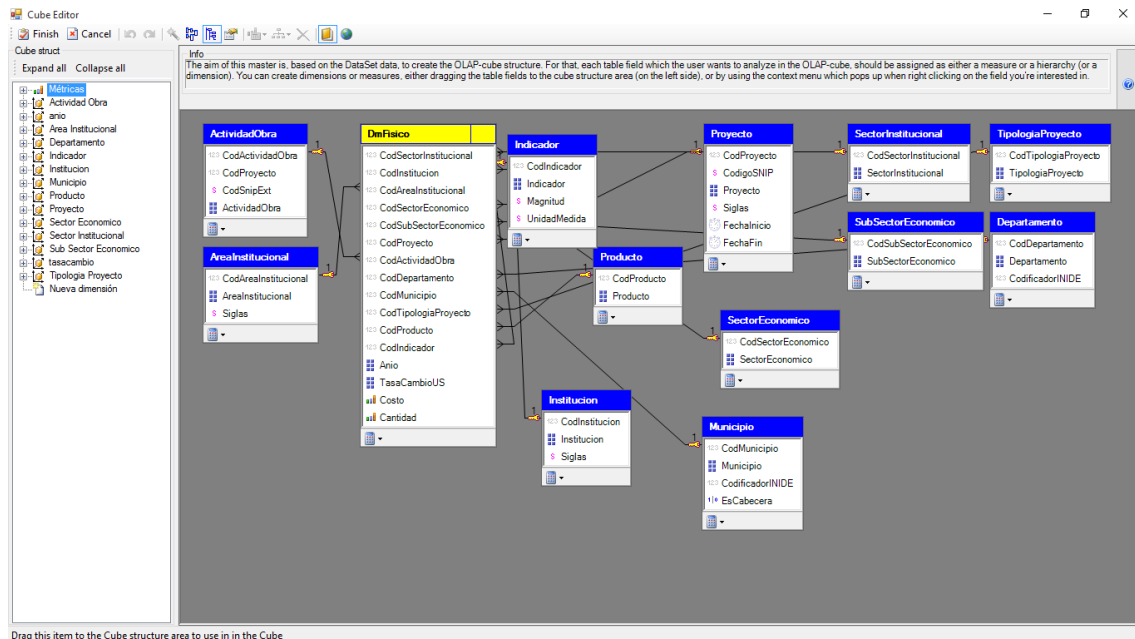
- Agregar a la forma web el componente RadarCube desde el cuadro de herramientas.

Ilustración 14: Componente RadarCube



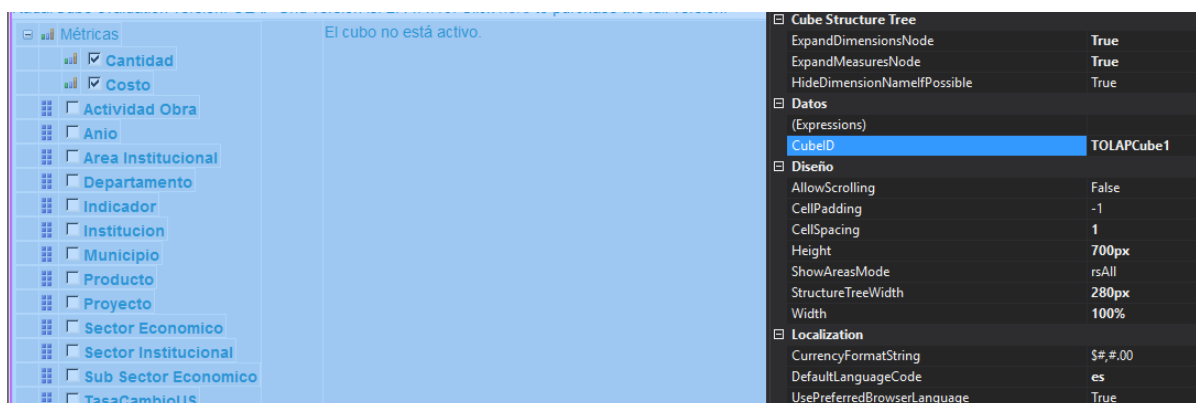
- Crear la estructura del cubo, mediante el DataSet creado como fuente de datos del cubo.
- Crear la estructura del cubo a partir del DataSet.

Ilustración 15: estructura del Cubo.



- Enlazar el cubo RadarCube con la tabla dinámica TolapGrid.
- Ejecutar la aplicación. El Grid se llena con los datos del DataSet.

Ilustración 16: TOLAPGrid enlazado con el RadarCube.



- Configurar el Radar Cube para cargarse de forma dinámica a fin de poder tener control de los datos. Esto se realizó empleando código directamente. En el anexo A se describe a detalle este proceso.

### 5.3.10.2 Implementación del Dashboard

El dashboard para el Sistema Nacional de Inversión Pública fue diseñado con el objetivo de ayudar en la tarea de visualizar los principales indicadores de la ejecución de la inversión pública. Entre ellos destaca el porcentaje de ejecución en los distintos proyectos que se llevan a cabo en el país.

- Visualizar el porcentaje de ejecución total en un año determinado.
- Visualizar el porcentaje de ejecución según:
  - Fondos propios
  - Fondos del tesoro
  - Donaciones
  - Presupuesto
- Visualizar el porcentaje de ejecución según institución.
- Visualizar el porcentaje de ejecución de un proyecto determinado.
- Imprimir los distintos gráficos y reportes presentados.

Para la implementación del cuadro de mando integral, se siguió el procedimiento descrito:

- Instalar High Charts en Visual Studio mediante el uso de Nuget.

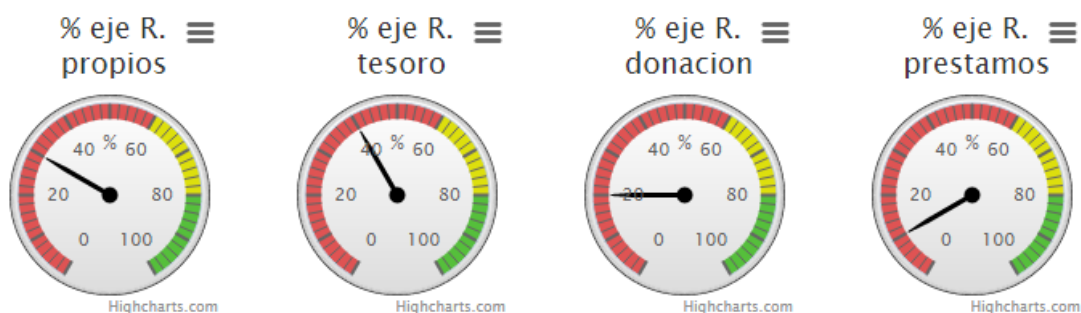
```
PM> Install-Package DotNet.Highcharts
```

- Agregar al proyecto, referencias a la suite de gráficas High Charts.
  - Añadir la referencia de la librerías JavaScript jQuery y HighCharts en la master page del proyecto
  - Crear los gráficos en la página, según el diseño acordado con el cliente.
- El código a continuación ejemplifica cómo se realiza este proceso en ASP.Net

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">  
    <asp:Literal ID="ltrChart" runat="server"></asp:Literal>  
</asp:Content>
```

- Configurar las propiedades de datos y de apariencia de cada Chart.
- Ejecutar la aplicación mostrando los indicadores.

Ilustración 17: Charts de tacómetro configuradas con los indicadores de avance del SNIP



Los detalles de este procedimiento se presentan en el Anexo A.

Cabe mencionar que ambas herramientas de BI se integraron en una misma aplicación Web para facilitar su uso y mantenimiento.

### 5.3.11 Mantenimiento y crecimiento

Para el crecimiento de los Data Marts, se definieron scripts que se cargaron como trabajos de base de datos que son ejecutados por el agente de MS SQL Server diariamente. Estos se pueden ejecutar a demanda si se requiere actualizar inmediatamente antes de ejecutar el analítico.

Para el mantenimiento de los Data Marts, en el evento de nuevos hechos o dimensiones, se requerirá la ampliación del modelo de datos, los scripts de ETL y las definiciones de los cubos en el analítico.

### 5.3.12 Implantación de la solución

La herramienta de Inteligencia de negocios se implantó como una aplicación Web en el servidor de aplicaciones de DGIP a lo interno de la institución, pero puede ser accedida desde el Internet a través de un URL temporal: <http://ws.snip.gob.ni/BPBI/>

Ilustración 18: Herramienta de BI publicada en el portal del SNIP



### 5.3.12.1 Creación del Data Warehouse en producción

Para la realización de este proceso, fue necesario primeramente la creación de la base de datos SnipDM que conforma el DatawareHouse implementado con tres data marts: Indicadores Físicos, Financiero anual y Financiero mensualizado. Esto se realizó con un script de creación de la estructura y posteriormente con un script para ETL que es el que se empleará para el mantenimiento. Se creó un usuario específico para el acceso de la aplicación a la base de datos y se concedieron los privilegios requeridos en la administración de la seguridad de SQL Server.

Se crearon los procedimientos almacenados de ejecución de las operaciones de ETL a partir de los scripts para el mantenimiento de la aplicación y se programaron tareas (jobs) con ejecución automática, inicialmente por la noche. De acuerdo a la necesidad de actualización de la información en el almacén de datos, se estará ajustando la periodicidad de ejecución de dichos paquetes ETL.

### **5.3.12.2 Creación de la aplicación web en el servidor de aplicaciones**

Para la creación de la aplicación web, el proyecto de visual Studio se publicó, incluyendo la copia local de las librerías de terceros a fin que todas las dependencias se encontraran integradas al paquete de instalación. Se copió el paquete y se creó la aplicación desde la consola de administración del IIS 7.0 del servidor de aplicaciones.

Fue requerido realizar los ajustes en el *connect string* de la aplicación a fin de redirigirla a la nueva base de datos con las credenciales creadas.

Asimismo fue requerido ajustar la seguridad de la aplicación para que pudiera publicarse, por cuanto el desarrollo se realizó con seguridad integrada (Windows) y el despliegue, de acuerdo a las políticas de Dirección de Desarrollo del Banco de Proyectos (DDBP), debe ser con seguridad por autenticación.

### **5.3.12.3 Inducción a usuarios y puesta en producción**

De acuerdo a lo establecido por DGIP, luego de la publicación de la herramienta, se realizó una inducción a cinco usuarios del uso de la herramienta. Los usuarios, al ser expertos en el negocio, fácilmente adquirieron dominio del uso de la herramienta y realizaron diferentes pruebas de funcionamiento.

La herramienta quedó publicada y disponible para su uso a lo interno de la institución, la Dirección de Desarrollo del Banco de Proyectos (DDBP) estará integrando la aplicación a su marco de trabajo de seguridad a fin de prepararla para optimizar su uso por roles de usuario y niveles de gestión.

### **5.3.13 Resultados de la aplicación de la solución**

Los usuarios que conformaron la primera fase de transferencia tecnológica de la herramienta fueron asignados con el propósito de evaluar la aplicación. La percepción del usuario respecto a la apariencia y facilidad de uso de la herramienta fue muy positivo y el desempeño de las consultas, a pesar de



estarse usando cubos en memoria. Los detalles de esta evaluación se reportan en la sección 5.4, Evaluación de Resultados.

#### 5.3.14 Resultados en la bitácora del Backlog

El proceso Scrum, tiene el backlog como instrumento central para organizar el flujo de trabajo. Al inicio del proceso se identificó el backlog del proyecto y éste se fue refinando a medida que avanzaban los sprints. Al final del proyecto, la bitácora de actualización del backlog ofrece una fotografía de cómo devino el proyecto, por lo cual se tiene a bien agregarla:

Tareas ejecutadas	Seguimiento	Revisión
<b>Planificación del proyecto</b>	Se realizaron reuniones de trabajo para definir los requerimientos del negocio.	Implementado Sprint0, se realizó exitosamente. Fue necesario realizar varios encuentros para los entregables de proyecto, según los requerimientos del negocio.
<b>Definición de los requerimientos del sistema</b>	Implementar las funcionalidades necesarias para gestionar información de entradas y salidas del proceso.	Implementado Sprint1, la forma en que se implantaron las funcionalidades cumplen con los objetivos
<b>Modelado dimensional</b>	Realizar una nueva arquitectura de la base de datos original.	Implementado Sprint 2, se realizó exitosamente ya que se logró erradicar la recursividad entre tablas mediante el modelo estrella.
<b>Diseño y desarrollo de la presentación de datos</b>	Desarrollar el proceso de ETL (Extracción, Transformación y carga de los Datos) mediante	Implementado Sprint 3, se definieron tablas de hechos y dimensiones para facilitar el proceso de transformación y carga de los datos.

Tareas ejecutadas		Seguimiento	Revisión
		el gestor de Base de Datos SQL Server.	
<b>Selección de productos e instalación</b>		Familiarización con las herramientas Syncfusion y Radarsoft.	Implementado Sprint 4, Se observó que ambas herramientas poseen limitaciones en cuanto a documentación para el desarrollo de OLAP.
		Pruebas de la herramienta, para elaboración OLAP.	Se realizaron pruebas exitosas. Syncfusion requiere un proveedor OLAP para sus controles BI.
		Evaluación de los resultados de las pruebas y presentación al cliente.	Se presentaron los resultados de las pruebas tanto empleando servicios de análisis como directamente en memoria.
		Elaboración de propuesta sobre la utilización de Análisis Server	Dado que los mejores resultados son con empleo de servicios de análisis en el gestor de Base de Datos se propuso al cliente estudiar esta posibilidad.
		Documentación sobre la herramienta Radarsoft y pruebas de la misma.	El cliente expreso su preferencia por la herramienta Radarsoft para la creación del OLAP mediante uso directo en memoria.  Se observó que la herramienta puede realizar la carga de datos desde un DataSet al cubo multidimensional.  El modelo directo, es más lento e ineficiente que el modelo con servicios de análisis de SQL Server.

Tareas ejecutadas	Seguimiento	Revisión
<b>Diseño del almacén de datos</b>	<p>Realizar análisis sobre la base de datos del SNIP para la extracción de tablas de hechos y dimensiones.</p> <p>Realización del proceso inicial de extracción y carga de datos (ETL).</p> <p>Implementación del analítico del Data Warehouse en la herramienta Radarsoft</p>	<p>Sprint 5 y 6 finalizado.</p> <p>Se realizaron varios encuentros con los involucrados para definir el modelo y las áreas más importantes para la realización de los Datawarehouse.</p> <p>Se implementaron dos datamarts que conforman el Data Warehouse.</p> <p>Se conectó el control de Radarsoft a la BD y se implementó el analítico.</p>
<b>Investigación sobre herramientas de software para la elaboración del dashboard:</b>	<p>Familiarización con las herramienta Syncfusion y Dundas.</p> <p>Evaluación sobre las ventajas y desventajas de las mismas</p> <p>Pruebas de la herramienta, para elaboración dashboard.</p>	<p>Sprint 7 finalizado.</p> <p>Syncfusion cuenta con poca documentación de sus controles.</p> <p>Dundas charts es una herramienta proporcionada por el SNIP la cual cuenta con una licencia propietaria pero es una versión vieja de la herramienta.</p> <p>Se realizaron las pruebas, con resultados ambivalentes.</p>
	<p>Familiarización con la herramienta Highchart.</p> <p>Evaluación sobre las ventajas y desventajas de la misma</p>	<p>Hlghchart cuenta con gran documentación de sus gráficos y ejemplos que facilitan su uso.</p>

Tareas ejecutadas	Seguimiento	Revisión
	Pruebas de la herramienta, para elaboración dashboard	DotNet highchart provee una alternativa para crear los gráficos desde el código en c# de cada web form, (originalmente los gráficos se crean en jquery), esto facilita su implementación en el proyecto
<b>Desarrollo del dashboard</b>	Diseño y desarrollo del dashboard según los requerimientos de la institución.	Se utilizó el estilo la plantilla de html y la hoja de estilos del SNIP la cual fue proporcionada por la institución, para los gráficos se utilizó highcharts ya que presenta los controles necesarios y demandados por la institución.
<b>Especificación de aplicaciones para usuarios finales.</b>	Seleccionar las herramientas de inteligencia de negocios idóneas para cada nivel de toma de decisiones.	El Spint 8, se completó correctamente. Nivel operativo: Reportes dinámicos. Nivel táctico: tablas dinámicas. Nivel Estratégico: herramientas graficas de visualización de datos.
<b>Mantenimiento y crecimiento del Data Warehouse</b>	Elaboración de script para alimentar la Data	Sprint 9, Se realizó correctamente.  Se desarrollaron scripts que se instalaron como Jobs programados en el Gestor de Bases de Datos para ejecutarse automáticamente para el mantenimiento del Data Warehouse. Estos corresponden al proceso de ETL.

Tareas ejecutadas	Seguimiento	Revisión
		El crecimiento conceptual (incluir más data marts) dependerá de las necesidades y los requerimientos del negocio.
<b>Evaluación</b>	<p>Realizar casos de prueba que realicen los usuarios y permitan comparar la pertinencia y desempeño de las herramientas respecto a las herramientas actuales con que cuentan.</p> <p>Consultar la percepción del usuario y consolidar resultados.</p>	Sprint 10 completado.

## 5.4 Evaluación de los resultados

Para poder evaluar los resultados de la aplicación de la herramienta de Inteligencia de Negocios y poder verificar si cumple con los objetivos planteados para DGIP, se analizaron sus factores de éxito a partir de los objetivos y la justificación, encontrándose que para ser exitosa debe:

- Requerir menos reportes
- Requerir menos tiempo para realizar la consulta
- Ser más fácil de usar
- Ser interactiva

- Proveer información visual más fácil de entender que la alfanumérica.

#### **5.4.1 Definición de las métricas**

A partir de analizar la naturaleza de los factores de éxito y agruparlos de acuerdo a éstas, se determinaron las métricas y medidas a utilizar:

##### **1. Desempeño:**

- 1.1. Tiempo de respuesta
- 1.2. Numero de reportes

##### **2. Facilidad de uso (usabilidad):**

- 2.1. Interfaz simple e intuitiva
- 2.2. Es Interactiva
- 2.3. Tiene información visual

Las medidas de desempeño se obtuvieron a través de la aplicación de un test comparativo con las herramientas actuales (reportes tabulares) y la herramienta de inteligencia de negocios desarrollada, en el cual se cronometró el tiempo y se contaron los números de reportes (o consultas) diferentes para obtener la información requerida.

Las medidas de usabilidad se obtuvieron mediante instrumento de encuesta a los testers de su percepción respecto al uso de las herramientas.

#### **5.4.2 Definición de instrumentos**

Se diseñó un test que tuvo como carga de trabajo la obtención de información histórica y actual a diferentes niveles de detalle y cruces de variables seleccionados:

- **Caso de prueba No. 1:** Verificar información de la programación vs la ejecución de un año por Institución, por Institución y proyecto, por Institución, Proyecto y Fuente de Financiamiento.

- **Caso de prueba No. 2:** Verificar información de la programación vs la ejecución de un año por Institución, por Sector Económico, por Departamento. Agregar por Subsector y por Municipio.
- **Caso de prueba No. 3:** Verificar información de la ejecución de indicadores físicos en un año por Institución, por Sector Económico, por Departamento y Tipología de Proyecto. Agregar un segundo año.
- **Caso de prueba No. 4:** Consultar el estado de los indicadores de Porcentaje de ejecución del Programa de Inversión Pública anual, tanto a nivel global como por tipo de recurso (financiamiento).

La encuesta consultó para cada herramienta una única respuesta para cada medida por tester, de acuerdo a las siguientes escalas de valores:

Medida de evaluación	Escalas de valores
<b>Interfaz simple e intuitiva</b>	Simple, Compleja
<b>Es Interactiva</b>	Sí, No
<b>Tiene información visual</b>	Sí, No

### 5.4.3 Resultados de la evaluación

A continuación se presentan los resultados de ambas evaluaciones: Test para comparar el desempeño de las herramientas y Encuesta para evaluar la percepción sobre la usabilidad de la nueva herramienta en comparación con la anterior.

#### 5.4.3.1 Resultados del Test

Se aplicó el test por parte de funcionarios de nivel táctico del órgano rector del SNIP. El equipo de testers consistió en cinco funcionarios, realizando los casos de prueba tanto con la herramienta anterior como con la nueva.

#### Resultado del Caso de Prueba 1.

En este caso de prueba, que consiste en el mejor escenario, por el orden de las consultas (drill down) no se requerían múltiples reportes para realizar la consulta con la herramienta anterior pues la capacidad de agrupación por nivel de los reportes puede suplir la funcionalidad. Sin embargo, la interactividad provista por la tabla dinámica de la herramienta de BI permitió a los ejecutores consultar la información en menor tiempo.

En este caso, se obtuvo un ahorro de tiempo promedio del 33%. No hubo ahorro en cantidad de pasos procedimentales.

La tabla a continuación detalla los tiempos resultado del caso de prueba 1, reportados en por los diferentes testers. La fórmula a continuación muestra el procedimiento definido para calcular el ahorro porcentual en cada caso:

$$Ahorro = \frac{(\text{Resultado}_{\text{metodo anterior}} - \text{Resultado}_{BI})}{\text{Resultado}_{\text{metodo anterior}}} * 100\%$$

C1	Herramienta Anterior (A)		Herramienta BI (B)		Ahorro (A - B)/A*100%	
	Tiempo de Respuesta (s)	# de Reportes	Tiempo de Respuesta (s)	# de Reportes	Tiempo (%)	# de reportes (%)
1	286	1	187	1	35%	0%
2	301	1	199	1	34%	0%
3	256	1	179	1	30%	0%
4	277	1	187	1	32%	0%
5	294	1	194	1	34%	0%
<b>Media</b>	<b>283</b>	<b>1</b>	<b>189</b>	<b>1</b>	<b>33%</b>	<b>0%</b>

## Resultado del Caso de Prueba 2.

En este caso de prueba, que consiste en el peor escenario, dado que al ser múltiples dimensiones a consultar, se obliga a la generación de múltiples reportes con la herramienta anterior. Las dimensiones que se agregan, no incrementan el número de reportes pero sí retrasan el proceso de obtención de resultados.



En este caso, se obtuvo un ahorro de tiempo promedio del 54%. Asimismo se ganó eficiencia en un 67% de ahorro en cantidad de pasos procedimentales.

<b>C2</b>	Herramienta Anterior (A)		Herramienta BI (B)		Ahorro (A - B)/A*100%	
<b>Tester</b>	Tiempo de Respuesta (s)	# de Reportes	Tiempo de Respuesta (s)	# de Reportes	Tiempo (%)	# de reportes (%)
1	564	3	282	1	50%	67%
2	593	3	266	1	55%	67%
3	532	3	245	1	54%	67%
4	597	3	271	1	55%	67%
5	578	3	263	1	54%	67%
<b>Media</b>	<b>573</b>	<b>3</b>	<b>265</b>	<b>1</b>	<b>54%</b>	<b>67%</b>

### Resultado del Caso de Prueba 3.

En este caso de prueba se emplea el cubo de Indicadores Físicos, a diferencia de los dos anteriores que emplean el cubo Financiero anual. La necesidad de agregar un año en este caso tiene el mismo impacto en ambas porque obliga a la herramienta BI volver a cargar la información. Podemos considerar este caso de prueba como uno de complejidad media.

En este caso, se obtuvo un ahorro de tiempo promedio del 40%. Se obtuvo un ahorro de 50% en cantidad de pasos procedimentales.

<b>C3</b>	Herramienta Anterior (A)		Herramienta BI (B)		Ahorro (A - B)/A*100%	
<b>Tester</b>	Tiempo de Respuesta (s)	# de Reportes	Tiempo de Respuesta (s)	# de Reportes	Tiempo (%)	# de reportes (%)
1	387	4	220	2	43%	50%
2	394	4	234	2	41%	50%
3	399	4	245	2	39%	50%
4	385	4	235	2	39%	50%
5	390	4	241	2	38%	50%
<b>Media</b>	<b>391</b>	<b>4</b>	<b>235</b>	<b>2</b>	<b>40%</b>	<b>50%</b>

### Resultado del Caso de Prueba 4.

En este caso de prueba se emplea el Dashboard en la herramienta de BI. En el procedimiento anterior se obliga a realizar cálculos posteriores a partir del reporte para obtener la medida de los indicadores. En la nueva herramienta, los indicadores se muestran desde su página de inicio.

En este caso, se obtuvo un ahorro de tiempo promedio del 97%. Se obtuvo un ahorro de 50% en cantidad de pasos procedimentales.

<b>C4</b>	Herramienta Anterior (A)		Herramienta BI (B)		Ahorro (A - B)/A*100%	
<b>Tester</b>	Tiempo de Respuesta (s)	# de Reportes	Tiempo de Respuesta (s)	# de Reportes	Tiempo (%)	# de reportes (%)
1	96	2	3	1	97%	50%
2	105	2	4	1	96%	50%
3	97	2	3	1	97%	50%
4	95	2	3	1	97%	50%
5	98	2	3	1	97%	50%
<b>Media</b>	<b>98</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>97%</b>	<b>50%</b>

Tomando en consideración que los resultados del tiempo de respuesta en los casos de prueba son consistentes entre sí al mostrar el mismo comportamiento entre testers y entre pruebas, se pueden promediar los resultados, concluyendo que en las pruebas, la herramienta de BI permite un ahorro de tiempo promedio de 42%.

Asimismo podemos concluir, aunque solo en un caso se obtuvo ahorro por evitar la necesidad de múltiples reportes, que en dependencia de la consulta, la herramienta BI reduce la necesidad de requerir generar múltiples reportes.

#### 5.4.3.2 Resultados de la Encuesta

Se aplicó la encuesta a los funcionarios testers. Esta consulta consistió en tres preguntas (medidas de usabilidad) que se preguntaron tanto para el método anterior como para la nueva herramienta, obteniéndose consenso que el método anterior es complejo y la herramienta de BI desarrollada es más simple e intuitiva y por ende, más fácil de usar. Asimismo, mientras el método anterior carece de

interactividad para el análisis y ayudas visuales, la nueva herramienta BI sí lo tiene.

Encuesta	Interfaz simple e intuitiva		Es Interactiva		Tiene información visual	
Tester	Método Anterior	Herramienta BI	Método Anterior	Herramienta BI	Método Anterior	Herramienta BI
1	Compleja	Simple	NO	SI	NO	SI
2	Compleja	Simple	NO	SI	NO	SI
3	Compleja	Simple	NO	SI	NO	SI
4	Compleja	Simple	NO	SI	NO	SI
5	Compleja	Simple	NO	SI	NO	SI
<b>Moda</b>	<b>Compleja</b>	<b>Simple</b>	<b>NO</b>	<b>SI</b>	<b>NO</b>	<b>SI</b>

En conclusión, los resultados de la evaluación evidencian que la herramienta de Inteligencia de Negocios implementada, va a proveer todos los beneficios esperados a los usuarios de la DGIP, lográndose así los objetivos del proyecto exitosamente.

## 5.5 Costo de la aplicación

### 5.5.1 Costo de hardware

La DGIP cuenta con sus propios servidores de aplicaciones y de base de datos. A continuación se mencionan los costos de los dispositivos de hardware que se ocuparon en el desarrollo de este proyecto.

- 1 Servidor de base de datos USD 12,000.00
- 1 Servidor de Aplicaciones USD 12,000.00
- 2 Computadoras de desarrollo USD 1,200.00

---

TOTAL USD 25,200.00

### 5.5.2 Costos de Softwares de desarrollo

La DGIP cuenta actualmente con todas las licencias para el desarrollo de la aplicación. A continuación se mencionan los costos de cada herramienta que se ocupó para este proyecto.

- Visual studio 2013 Professional USD 500.00
- RadarSoft USD 1,250.00
- SqlServer 2008 USD 5,000.00

• HighChart	USD	0.00
<hr/>		
TOTAL	USD	6,750.00

### 5.5.3 Costo de Recuso Humano en el desarrollo de la herramienta

El Proyecto cuenta con dos desarrolladores los cuales trabajaron 300 horas laborales con un costo por hora de USD 7.00.

• Desarrollador 1	300 horas X USD 7.00	USD 2,100.00
• Desarrollador 2	300 horas X USD 7.00	USD 2,100.00
<hr/>		
TOTAL	USD	4,200.00

El costo de total de la inversión en el desarrollo e implementación de la herramienta es de USD 36,150.00

## 6 CONCLUSIONES Y RECOMENDACIONES

### 6.1.1 Conclusiones

- Se logró asimilar los procesos de Negocio del SNIP, lo que permitió diseñar una propuesta orientada a inteligencia de negocios que faciliten el monitoreo, seguimiento y evaluación de la ejecución de la Inversión Pública Nacional.
- La Metodología de Kimball resultó ser muy completa y apropiada para guiar el proceso de desarrollo de software de Sistemas de Información de Inteligencia de Negocios, como la herramienta desarrollada en el presente trabajo. La incorporación a apoyos que proporcionó el marco de trabajo de la metodología de desarrollo ágil de software Scrum, funcionó a la perfección.
- La utilización de la herramienta Radarsoft permitió la construcción de un cubo OLAP en memoria con generación de tablas dinámicas mediante

cruces de datos, partiendo de una base de datos transaccional facilitando el análisis y evaluación de la ejecución de la inversión pública.

- La utilización de la herramienta Highcharts permitió obtener una visualización atractiva a los usuarios del SNIP mediante indicadores de toma de decisión logrando resultados positivos en la presentación de los datos a través de reportes dinámicos y tableros.
- La solución de inteligencia de negocios desarrollada para el Sistema Nacional de Inversión Pública cumple con los requerimientos de los usuarios facilitando los procesos en la toma de decisiones a nivel estratégico y táctico, asegurando que cada grupo operativo tenga acceso a la información necesaria para contestar preguntas específicas y distribuir dicha información a todos los niveles de la organización.
- La evaluación realizada de la aplicación de la herramienta permitió verificar que la solución propuesta es adecuada y pertinente para los objetivos propuestos, permitiendo importantes ahorros de tiempo en los procesos de obtención y análisis de información, además de ser más fácil de usar que el método anterior.

### **6.1.2 Recomendaciones**

- Sobre la base de los resultados de la evaluación interna realizada a la herramienta Radar Cube de RadarSoft, se recomienda a la Dirección de Desarrollo del Banco de Proyectos, de la DGIP, considerar la migración del modo directo (cubo en memoria) al modo basado en servicios de análisis de SQL Server, con el cual se lograría acelerar el rendimiento de la aplicación en al menos un 30% (según las pruebas realizada por los desarrolladores al verificar el funcionamiento de ambos modos).



## 7 BIBLIOGRAFÍA

- Academia latinoamericana de business intelligence [ALBI]. (15 de 10 de 2010). Construyendo una Solución OLAP. *Construyendo una Solución OLAP*.
- Andersen, K. (31 de 12 de 2013). *The Importance of the Business Intelligence Dashboard*. Obtenido de The Importance of the Business Intelligence Dashboard: <http://www.mass.com/blog/bid/155733/The-Importance-of-the-Business-Intelligence-Dashboard>
- Anthony, R. N. (1965). *Planning and Control Systems. A framework for Analysis*.
- Araníbar, S. J. (2003). Inteligencia de Negocios. *Ciencia y Cultura*.
- Ayala, A. P. (2006). Inteligencia de Negocios: Una propuesta para su desarrollo en las organizaciones. México, México.
- Caralt, J., Curto, D., & Conesa, J. (2010). *Introducción al Business Intelligence*. Barcelona: UOC.
- Codd, E., Codd, S., & Salley, C. (1993). *Providing OLAP to user-analysts: An IT mandate*. Sunnyvale, CA: E. F. Codd & Associates.
- Escolar, F. (23 de 06 de 2013). *SCRUM con TFS*. Obtenido de Revista Digital "Programando con .NET": <http://programandonet.com/web/scrum-con-tfs/>
- Henessy, J., & Patterson, D. (1993). *Arquitectura de Computadoras, Un enfoque cuantitativo*. Madrid: McGraw Hill Latinoamericana.
- IDORU. (5 de 05 de 2012). *diseño de Bases de Datos*. Obtenido de <http://www.interaktiv.cl/blog/>: [http://www.interaktiv.cl/blog/wp-content/uploads/2012/04/4.-Metodologia\\_disegno\\_DW.pdf](http://www.interaktiv.cl/blog/wp-content/uploads/2012/04/4.-Metodologia_disegno_DW.pdf)
- Inmon, W. H. (2002). *Building the Data Warehouse*. New York: John Wiley & Sons, Inc.
- Kimball, R., & Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* (2 ed.). New York: John Wiley & Sons, Inc.
- Mendoza, R. (4 de Febrero de 2008). *Conceptos Inteligencia de Negocios - Business Intelligence*. Obtenido de Conceptos Inteligencia de Negocios - Business Intelligence: <http://rimenri.blogspot.com/>
- Morales, A. (2012). Inteligencia de negocios, una ventaja competitiva para las organizaciones. *Ciencia y Tecnología Vol. 8, núm. 22, 1 - 12*.
- Oracle. (22 de 12 de 2014). *Oracle business Intelligence Tutorial*. Obtenido de Oracle business Intelligence Tutorial: [http://docs.oracle.com/html/E10312\\_01/dm\\_concepts.htm](http://docs.oracle.com/html/E10312_01/dm_concepts.htm)
- Oracle. (22 de 12 de 2014). *Oracle Business Intelligence Tutorial*. Obtenido de Oracle Business Intelligence Tutorial: [http://docs.oracle.com/html/E10312\\_01/dm\\_concepts.htm](http://docs.oracle.com/html/E10312_01/dm_concepts.htm)
- proyectosagiles.org. (12 de 03 de 2015). *Cómo gestionar proyectos ágiles con SCRUM*. Obtenido de Qué es SCRUM: <http://www.proyectosagiles.org/que-es-scrum>

- Rivadera, G. (2010). *Metodología de Kimball para diseño de almacenes de datos (Datawarehouses)*. Buenos Aires: Facultad de Ingeniería e Informática, UCASAL.
- Rosado, A., & Rico, D. (Abril 2010). Inteligencia de Negocios: Estado del Arte. *Scientia Et Technica*, vol. XVI(44), 1 - 7.
- Rouse, M. (Noviembre de 2010). <http://searchbusinessanalytics.techtarget.com>. Obtenido de <http://searchbusinessanalytics.techtarget.com/definition/business-intelligence-dashboard>
- Shutherland, J., & Shwaber, K. (01 de 07 de 2013). *La guía de SCRUM*. Obtenido de SCRUM Guides: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Sinergia e Inteligencia de Negocio S.L. (10 de 12 de 2015). *Qué es Bussiness Intelligence*. Obtenido de SINNEXUS: Bussiness Intelligence - Información Estratégica: [http://www.sinnexus.com/business\\_intelligence/index.aspx](http://www.sinnexus.com/business_intelligence/index.aspx)
- The tingley advantage. (20 de 06 de 2013). *The tingley advantage*. Obtenido de The importance of Dashboards: <http://www.thetingleyadvantage.com/2013/06/the-importance-of-dashboards.html>
- Tiwari, P. (04 de 06 de 2014). *Importance of Dashboard in Business Intelligence in Qlikview*. Obtenido de Importance of Dashboard in Business Intelligence in Qlikview: <http://www.c-sharpcorner.com/UploadFile/35fef0/importance-of-dashboard-in-business-intelligence-in-qlikview/>



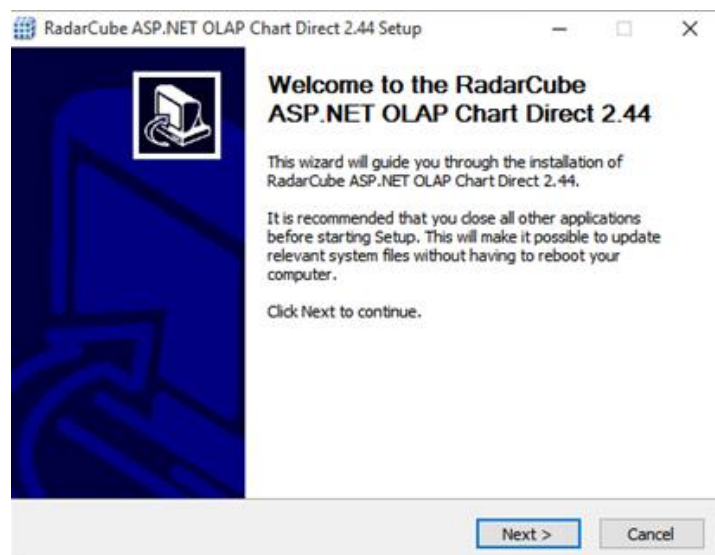
## 8 ANEXOS

### 8.1 Anexo A: Manual de referencia técnica de la Herramienta

#### 8.1.1 Instalación de Suite de controles OLAP de Radarsoft

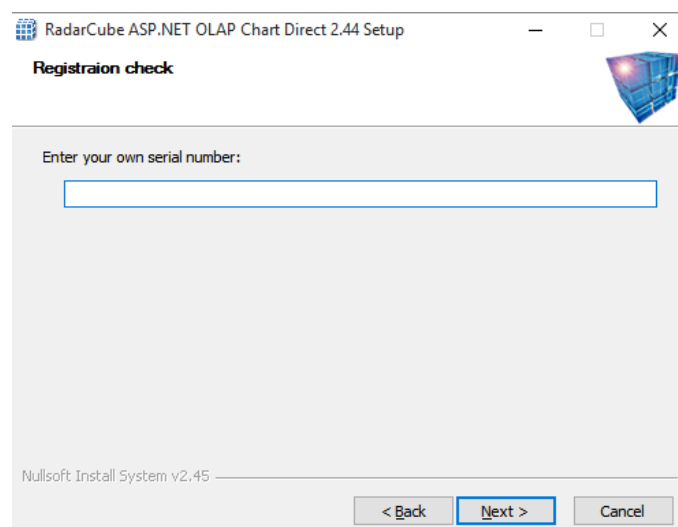
La instalación de la suite de controles OLAP de RadarSoft resulta ser muy sencilla ya que es guiada por un asistente.

***Ilustración 19: Inicio de instalación de RadarSoft***



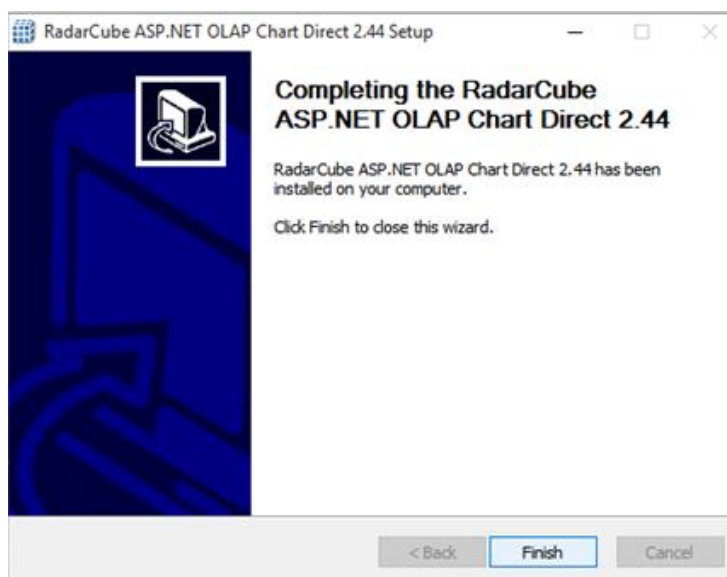
Este programa es propietario así que es necesario ser propietario de una licencia

***Ilustración 20: Radarsoft Licencia***



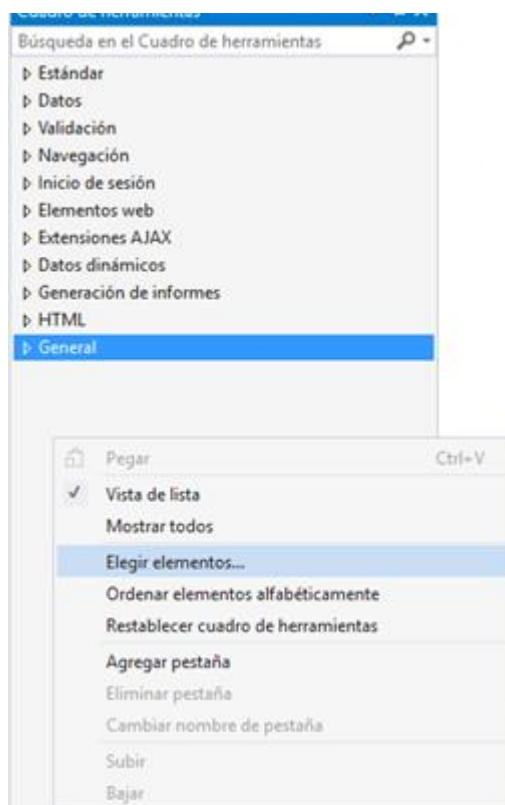
Una vez completada la instalación debemos hacer click en finalizar

### ***Ilustración 21: Radarsoft Instalación***



En el Visual Studio se deben habilitar los controles de RadarSoft en el panel de herramientas. Se debe abrir el cuadro de herramientas y mediante click derecho en éste se selecciona la opción elegir elementos.

### **Ilustración 22: Habilitar controles de Radarsoft**



Posteriormente se debe filtrar, por ejemplo con la palabra “Radar” y aparecerán los controles de RadarSoft, se recomienda seleccionar todos y dar click en aceptar.

### **8.1.2 Desarrollo de la Herramienta de BI**

A continuación se explica el proceso seguido para el desarrollo de la Herramienta de Inteligencia de Negocios que integra tanto el analítico como el dashboard. Esta guía de referencia se centra en los detalles técnicos que pudieran ser relevantes para el personal a cargo del mantenimiento y extensión de la herramienta.

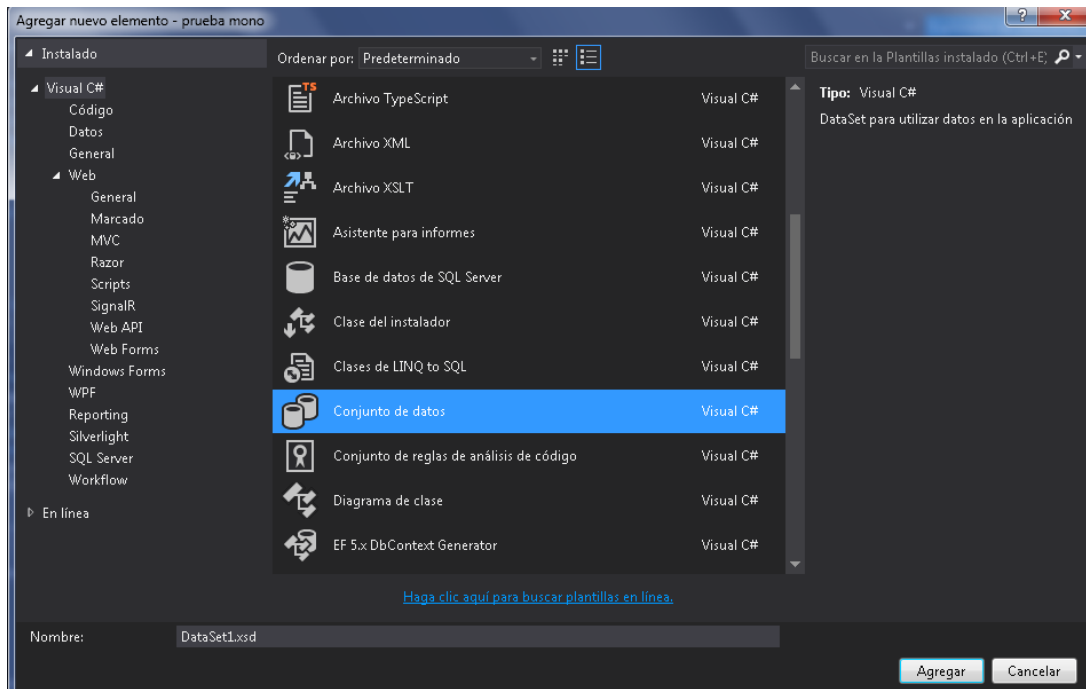
#### **Paso 1**

Creación del ETL para la elaboración de una Base de Datos totalmente desnormalizada con un modelo dimensional de tipo estrella que contendrá las tablas de hechos y dimensiones.

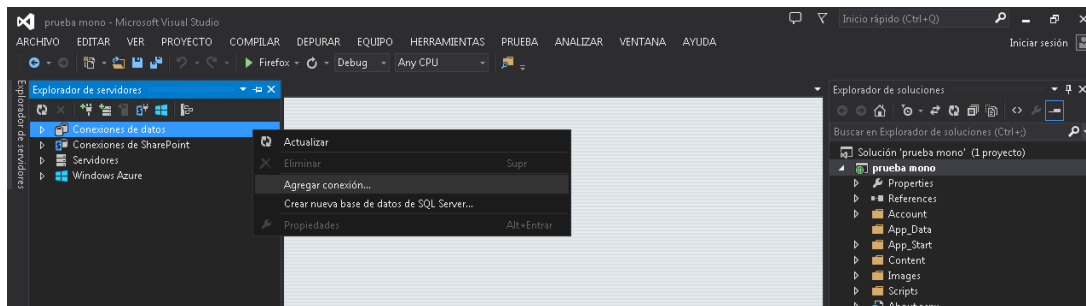
#### **Paso 2**

- Crear un nuevo proyecto web en visual estudio.
- Crear un DataSet y seleccionar la BD que fue creada desde el gestor de BD de SQL

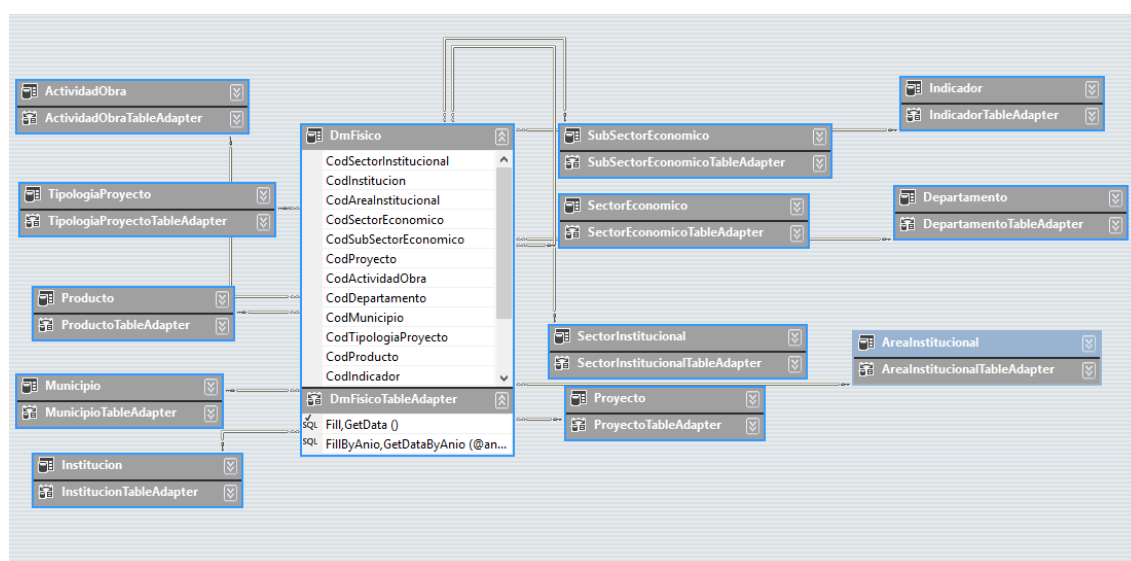
• **Ilustración 23: Creación del DataSet**



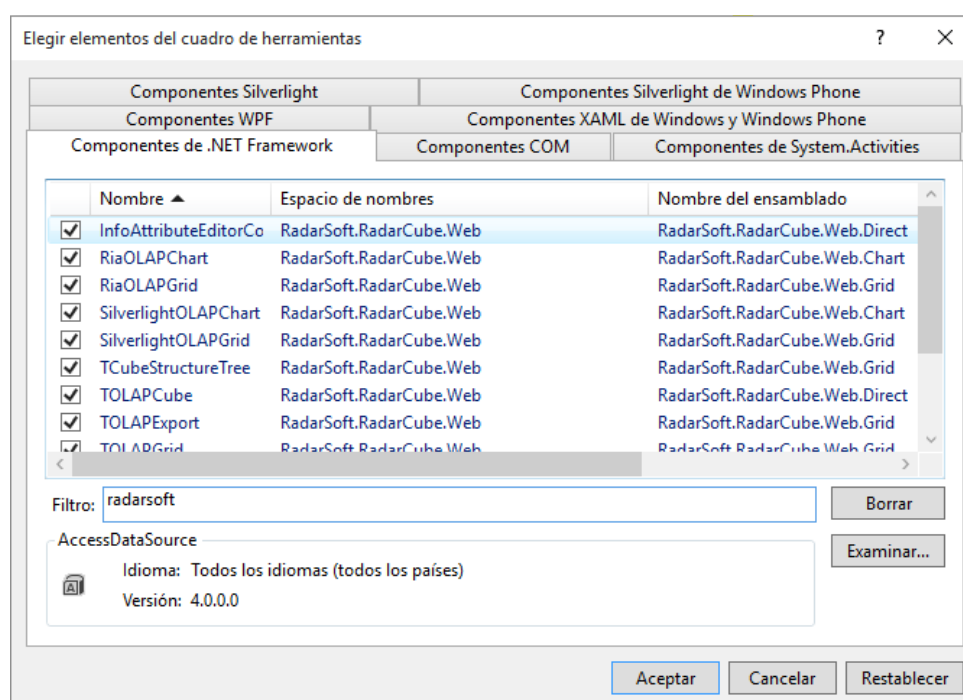
**Ilustración 24: Conexión de Datos**



**Ilustración 25: DataSet**

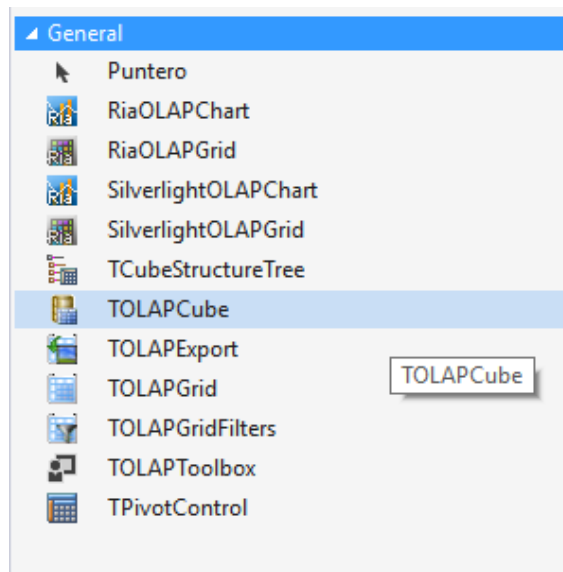


**Ilustración 26: Habilitar controles de Radarsoft**



Ya una vez hecho esto aparecerán las herramientas de radarsoft en nuestro panel de herramientas.

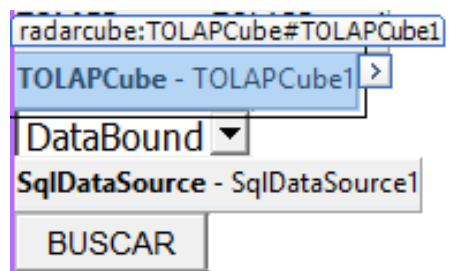
**Ilustración 27: Controles de Radarsoft**



#### Paso 5

Seleccionar la pestaña Diseño de nuestra página web y agregar el componente RadarCube desde el cuadro de herramientas.

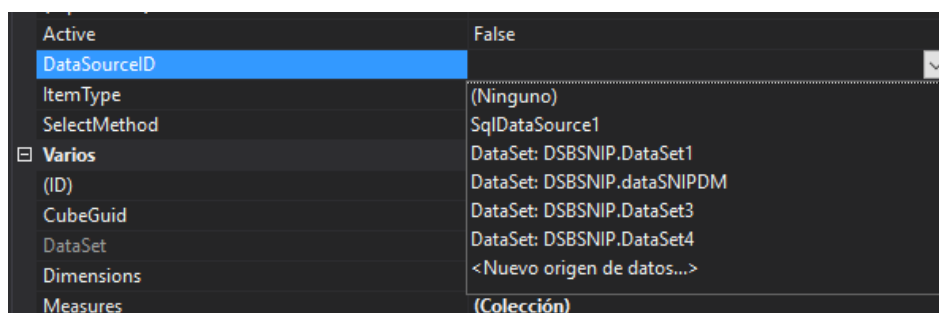
**Ilustración 28: Componente RadarCube**



#### Paso 6

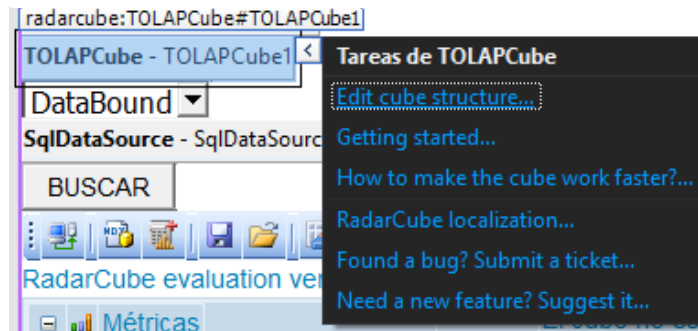
Crear la estructura del cubo, para esto se debe añadir el DataSet como la fuente de datos para nuestro cubo.

**Ilustración 29: DataSet como fuente de datos**



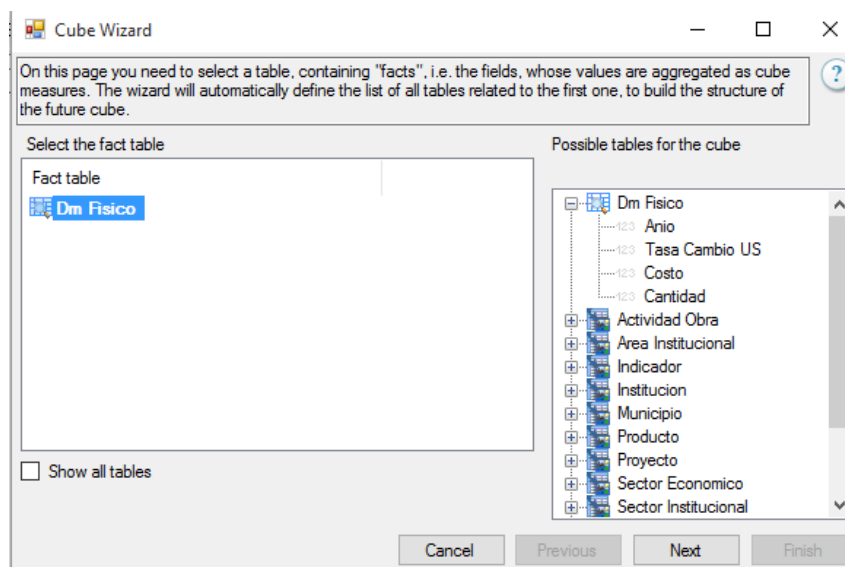
Luego iremos al cubo en el modo diseñador de la pantalla y damos click en el menú del cubo esto nos desplegará la opción crear estructura del cubo

### ***Ilustración 30: Estructura del Cubo***



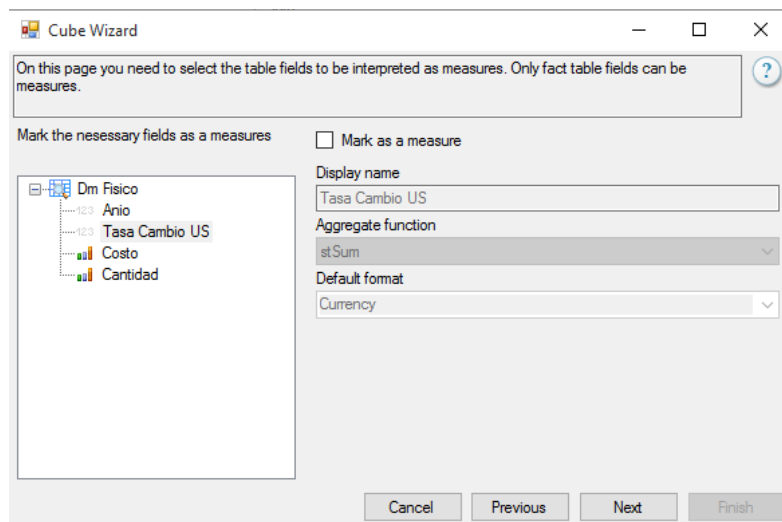
Al darle click en la opción nos abrirá una ventana que nos guiara paso a paso para crear la estructura

### ***Ilustración 31: Estructura del Cubo***



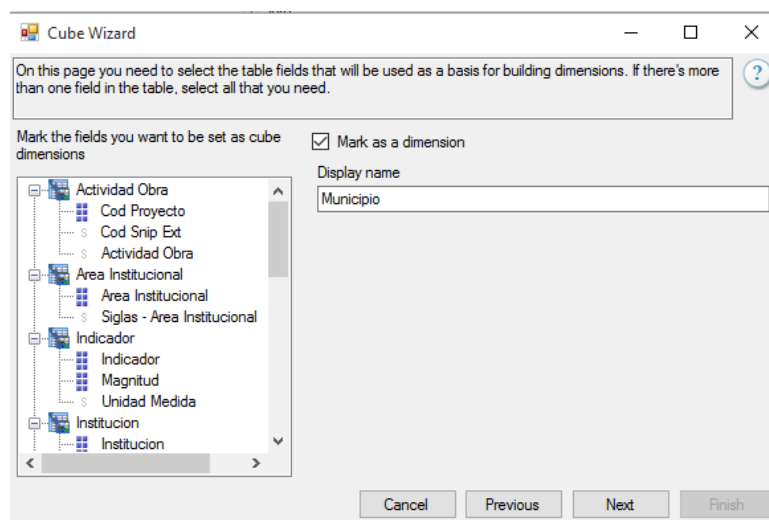
Damos click en el botón siguiente, y nos llevara a las opciones para elegir los campos que serán considerados como hechos, elegiremos en este caso los campos costo y cantidad

### ***Ilustración 32: Estructura del Cubo***



Al terminar de elegir los campos hechos vamos a las siguientes opciones que nos mostrara las tablas que contienen nuestras dimensiones, elegiremos los campos de las tablas que queremos mostrar.

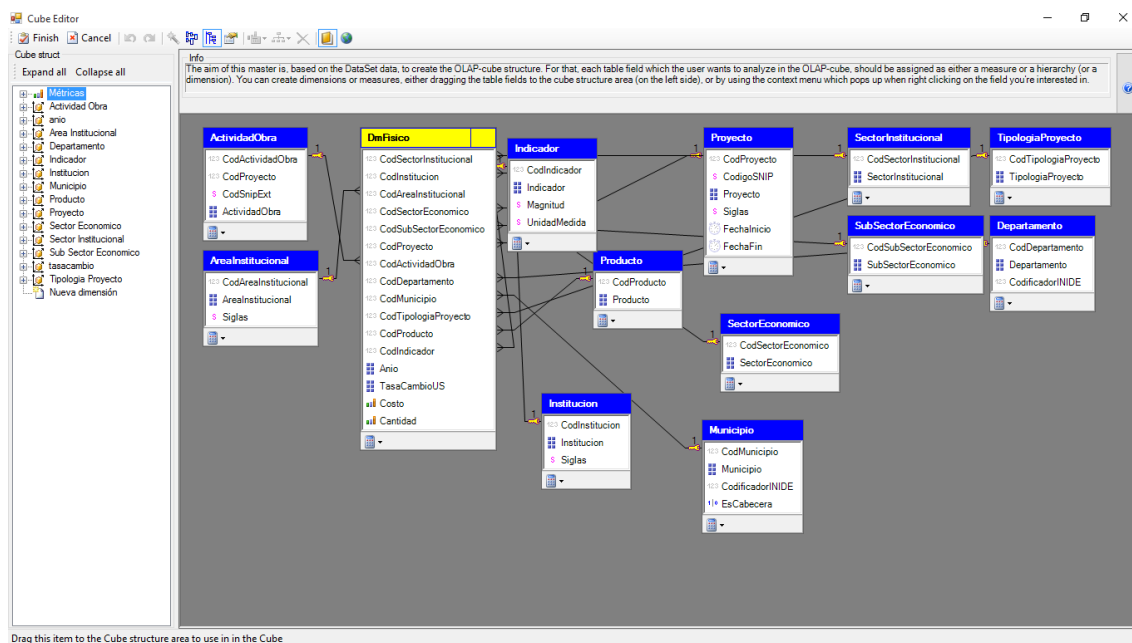
### ***Ilustración 33: Estructura del Cubo***





Elegiremos los campos necesarios y luego damos click en siguiente y luego terminar, esto nos mostrara la estructura del cubo que maneja Radar para crear los reportes.

### Ilustración 34: Estructura del Cubo



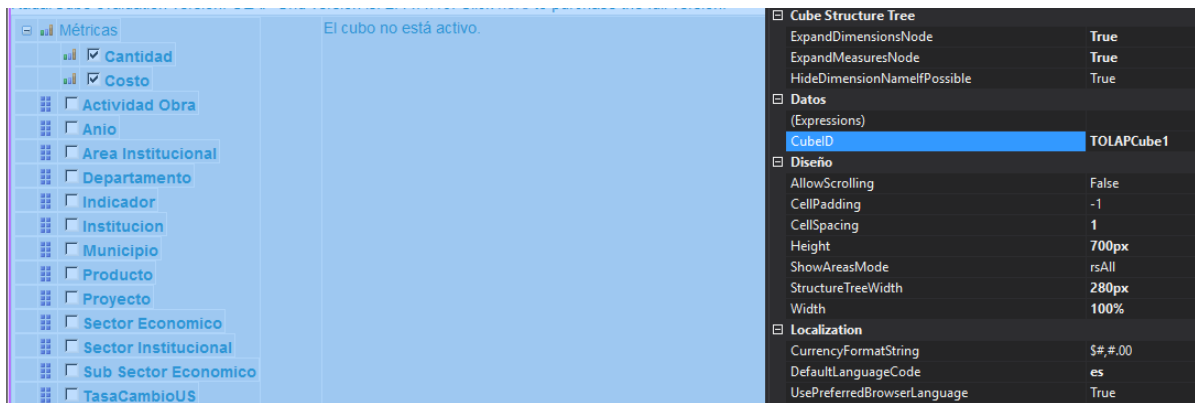
Aquí podremos seguir editando la estructura del cubo, si nos vamos a la vista de código podremos ver que se ha creado código html en la sección interna del control radarcube que corresponde a la estructura que hemos creado.

```
<radarcube:TOLAPCube ID="TOLAPCube1" runat="server" CubeGuid="e11a6a6-0000-0000-0000-000000000000" FactTableName="DmFisico" DataSourceID="DataSet: DS8SNIP.DataSet1">
  <Measures>
    <radarcube:TCubeMeasure DataTable="DmFisico" DisplayName="Costo" SourceField="Costo" SourceFieldType="System.Decimal" UniqueName="1cfa3ae9-b384-4d17-a948-14bb1ff2">
    <radarcube:TCubeMeasure DataTable="DmFisico" DisplayName="Cantidad" SourceField="Cantidad" SourceFieldType="System.Double" UniqueName="8c2c84e6-fcc1-4ee7-b59e-7da">
    </Measures>
  </Dimensions>...</Dimensions>
  <CubeTablePositions>...</CubeTablePositions>
  <TableAdapters>...</TableAdapters>
</radarcube:TOLAPCube>
```

### Paso 7

Ya una vez construida la estructura del cubo podemos enlazar el RadarCube con el TolapGrid

### Ilustración 35: RadarCube con TolapGrid



Una vez hecho esto nuestra página estará lista para ser ejecutada ya que el RadarCube se carga con los datos automáticamente del DataSet.

#### 8.1.3 Procedimiento para llenar el RadarCube

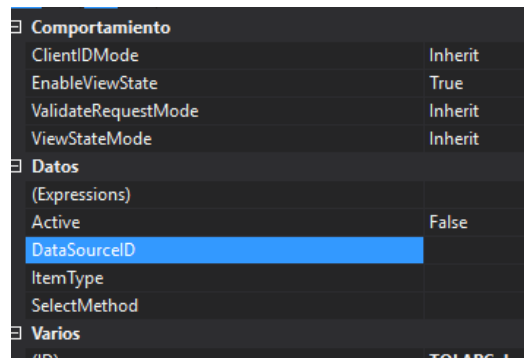
El RadarCube automáticamente obtendrá los datos del data set que se ha creado y establecido como fuente de datos, pero esto no nos permite controlar como mostraremos los datos.

A continuación se explicara la manera en que se deberá llenar el RadarCube de manera dinámica.

##### Paso 1

Quitaremos en las propiedades del RadarCube la fuente de datos y estableceremos como desactivado al cubo.

### Ilustración 36: RadarCube Desactivado



Esto nos permitirá controlar el cubo desde el código de C#, la estructura del cubo no se perderá.

#### Paso 2

En el ejemplo que se está desarrollando se controla los datos de cubo a través de un DropDownList que muestra datos del campo anio de la tabla Fuente.

Se creara una instancia del DataSet en el código de c#, luego crearemos un TableAdapter para cada una de las tablas presentes en el DataSet y los llenaremos desde el código, y la tabla Fuente será llenada dependiendo del anio que se ha establecido en el DropDownList

```
SqlConnection conexion= new SqlConnection(connectionString);
```

```
// AQUI SE ESTABLECEN LOS ANIOS QUE PRESENTARA EL GRID, CADA DATA  
ADAPTER SE LLENA CON UNA SENTENCIA DE SQL
```

```
SqlDataAdapter adpaterDMFisico = new SqlDataAdapter("SELECT Anio, Cantidad,  
CodActividadObra, CodAreaInstitucional, CodDepartamento, CodIndicador, CodInstitucion,  
CodMunicipio, CodProducto, CodProyecto, CodSectorEconomico, CodSectorInstitucional,  
CodSubSectorEconomico, CodTipologiaProyecto, Costo, TasaCambioUS FROM DmFisico  
WHERE (Anio = "+ Convert.ToString( DropDownList1.SelectedVale)+")", conexion);
```

```
SqlDataAdapter adpaterCodActividadObra = new SqlDataAdapter("SELECT CodActividadObra,  
CodProyecto, CodSnipExt, ActividadObra FROM dbo.ActividadObra", conexion);
```

```
SqlDataAdapter adpaterCodTipologiaProyecto = new SqlDataAdapter("SELECT  
CodTipologiaProyecto, TipologiaProyecto FROM dbo.TipologiaProyecto", conexion);
```

```
SqlDataAdapter adpaterCodSubSectorEconomico = new SqlDataAdapter("SELECT  
CodSubSectorEconomico, SubSectorEconomico FROM dbo.SubSectorEconomico", conexion);
```

```
SqlDataAdapter adpaterCodSectorInstitucional = new SqlDataAdapter("SELECT  
CodSectorInstitucional, SectorInstitucional FROM dbo.SectorInstitucional", conexion);
```

```
SqlDataAdapter adpaterCodSectorEconomico = new SqlDataAdapter("SELECT  
CodSectorEconomico, SectorEconomico FROM dbo.SectorEconomico", conexion);
```

```
SqlDataAdapter adpaterCodProyecto = new SqlDataAdapter("SELECT CodProyecto,  
CodigoSNIP, Proyecto, Siglas, FechaInicio, FechaFin FROM dbo.Proyecto", conexion);
```

```
SqlDataAdapter adpaterCodProducto = new SqlDataAdapter("SELECT CodProducto,  
Producto FROM dbo.Producto", conexion);
```

```
SqlDataAdapter adpaterCodMunicipio = new SqlDataAdapter("SELECT CodMunicipio,  
Municipio, CodificadorINIDE, EsCabecera FROM dbo.Municipio", conexion);
```

```
SqlDataAdapter adpaterCodInstitucion = new SqlDataAdapter("SELECT CodInstitucion,  
Institucion, Siglas FROM dbo.Institucion", conexion);
```

```
SqlDataAdapter adpaterCodIndicador = new SqlDataAdapter("SELECT CodIndicador,  
Indicador, Magnitud, UnidadMedida FROM dbo.Indicador", conexion);
```

```
SqlDataAdapter adpaterCodDepartamento = new SqlDataAdapter("SELECT  
CodDepartamento, Departamento, CodificadorINIDE FROM dbo.Departamento", conexion);
```

```
SqlDataAdapter adpaterCodAreaInstitucional = new SqlDataAdapter("SELECT  
CodAreaInstitucional, AreaInstitucional, Siglas FROM dbo.AreaInstitucional", conexion);
```

Luego añadimos los datos de cada table adapter a nuestra instancia del DataSet

```
adpaterCodActividadObra.Fill(dsTest, "ActividadObra");  
adpaterCodTipologiaProyecto.Fill(dsTest, "TipologiaProyecto");  
adpaterCodSubSectorEconomico.Fill(dsTest, "SubSectorEconomico");  
adpaterCodSectorInstitucional.Fill(dsTest, "SectorInstitucional");  
adpaterCodSectorEconomico.Fill(dsTest, "SectorEconomico");  
adpaterCodProyecto.Fill(dsTest, "Proyecto");  
adpaterCodProducto.Fill(dsTest, "Producto");  
adpaterCodMunicipio.Fill(dsTest, "Municipio");  
adpaterCodInstitucion.Fill(dsTest, "Institucion");  
adpaterCodIndicador.Fill(dsTest, "Indicador");  
adpaterCodDepartamento.Fill(dsTest, "Departamento");  
adpaterCodAreaInstitucional.Fill(dsTest, "AreaInstitucional");
```

```
adpaterDMFisico.Fill(dsTest, "DmFisico");
```

Luego agregamos nuestra instancia del DataSet como DataSource del cubo

```
TOLAPCube1.DataSet = dsTest;
```

```
TOLAPCube1.Active = true;
```

- **Ejemplo de estructura de un Cubo**

```
<radarcube:TOLAPCube ID="TOLAPCube1" runat="server" CubeGuid="ee11a6a6-0000-0000-0000-000000000000" FactTableName="DmFisico">
```

```
<Measures>
```

```
<radarcube:TCubeMeasure DataTable="DmFisico" DisplayName="Costo"
SourceField="Costo" SourceFieldType="System.Decimal" UniqueName="1cfa3ae9-b384-4d17-
a948-14bb1ff26903" Visible="True" />
```

```
<radarcube:TCubeMeasure DataTable="DmFisico" DisplayName="Cantidad"
SourceField="Cantidad" SourceFieldType="System.Double" UniqueName="8c2c84e6-fcc1-4ee7-
b59e-7daeac947b2c" Visible="True" />
```

```
</Measures>
```

```
<Dimensions>
```

```
<radarcube:TCubeDimension DisplayName="Actividad Obra" UniqueName="[Actividad
Obra]">
```

```
<Hierarchies>
```

```
<radarcube:TCubeHierarchy DataTable="ActividadObra"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="ActividadObra"
DisplayFieldType="System.String" DisplayName="Actividad Obra" IDField="ActividadObra"
IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total" UniqueName="895fd761-
37b1-4225-99b9-567f4de3376f" UnknownMemberName="Desconocido">
```

```
</radarcube:TCubeHierarchy>
```

```
</Hierarchies>
```

```
</radarcube:TCubeDimension>
```

```
<radarcube:TCubeDimension DisplayName="Area Institucional" UniqueName="[Area
Institucional]">
```

```
<Hierarchies>
```

```
<radarcube:TCubeHierarchy DataTable="AreaInstitucional"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="AreaInstitucional"
DisplayFieldType="System.String" DisplayName="Area Institucional" IDField="AreaInstitucional"
```

```

IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total"
UniqueName="767deded-20c7-4610-bc29-657e18275708"
UnknownMemberName="Desconocido">

    </radarcube:TCubeHierarchy>

</Hierarchies>

</radarcube:TCubeDimension>

<radarcube:TCubeDimension DisplayName="Indicador" UniqueName="[Indicador]">

    <Hierarchies>

        <radarcube:TCubeHierarchy DataTable="Indicador"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Indicador"
DisplayFieldType="System.String" DisplayName="Indicador" IDField="Indicador"
IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total"
UniqueName="466b5ed2-6f25-45dc-bc92-96da5614d231"
UnknownMemberName="Desconocido">

            </radarcube:TCubeHierarchy>

        </Hierarchies>

    </radarcube:TCubeDimension>

    <radarcube:TCubeDimension DisplayName="Institucion" UniqueName="[Institucion]">

        <Hierarchies>

            <radarcube:TCubeHierarchy DataTable="Institucion"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Institucion"
DisplayFieldType="System.String" DisplayName="Institucion" IDField="Institucion"
IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total"
UniqueName="86762055-538b-4831-a82c-eef7f074b61a"
UnknownMemberName="Desconocido">

                </radarcube:TCubeHierarchy>

            </Hierarchies>

        </radarcube:TCubeDimension>

        <radarcube:TCubeDimension DisplayName="Municipio" UniqueName="[Municipio]">

            <Hierarchies>

                <radarcube:TCubeHierarchy DataTable="Municipio"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Municipio"
DisplayFieldType="System.String" DisplayName="Municipio" IDField="Municipio"
IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total" UniqueName="25f860c8-
4676-4032-87f7-6dec820b4672" UnknownMemberName="Desconocido">

                    </radarcube:TCubeHierarchy>

                </Hierarchies>

            </radarcube:TCubeDimension>

        </Hierarchies>

    </radarcube:TCubeDimension>

</Hierarchies>

```

```

</radarcube:TCubeDimension>

<radarcube:TCubeDimension DisplayName="Producto" UniqueName="[Producto]">
  <Hierarchies>
    <radarcube:TCubeHierarchy DataTable="Producto"
      DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Producto"
      DisplayFieldType="System.String" DisplayName="Producto" IDField="Producto"
      IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total"
      UniqueName="17d3cb6e-27b5-4136-bcd8-a4ab0264fe9a"
      UnknownMemberName="Desconocido">

      </radarcube:TCubeHierarchy>
    </Hierarchies>
  </radarcube:TCubeDimension>

  <radarcube:TCubeDimension DisplayName="Proyecto" UniqueName="[Proyecto]">
    <Hierarchies>
      <radarcube:TCubeHierarchy DataTable="Proyecto"
        DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Proyecto"
        DisplayFieldType="System.String" DisplayName="Proyecto" IDField="Proyecto"
        IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total" UniqueName="4fe1c855-
        f3bb-4b5d-83a2-d77ffa36767" UnknownMemberName="Desconocido">

        </radarcube:TCubeHierarchy>
      </Hierarchies>
    </radarcube:TCubeDimension>

    <radarcube:TCubeDimension DisplayName="Sector Economico"
      UniqueName="[Sector Economico]">
      <Hierarchies>
        <radarcube:TCubeHierarchy DataTable="SectorEconomico"
          DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="SectorEconomico"
          DisplayFieldType="System.String" DisplayName="Sector Economico"
          IDField="SectorEconomico" IDFieldType="System.String" Origin="hoAttribute"
          TotalCaption="Total" UniqueName="0b1a19ae-d3f1-45ba-8754-ac01e99d490c"
          UnknownMemberName="Desconocido">

          </radarcube:TCubeHierarchy>
        </Hierarchies>
      </radarcube:TCubeDimension>

      <radarcube:TCubeDimension DisplayName="Sector Institucional"
        UniqueName="[Sector Institucional]">
        <Hierarchies>

```

```

        <radarcube:TCubeHierarchy DataTable="SectorInstitucional"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="SectorInstitucional"
DisplayFieldType="System.String" DisplayName="Sector Institucional"
IDField="SectorInstitucional" IDFieldType="System.String" Origin="hoAttribute"
TotalCaption="Total" UniqueName="3368a8f6-e398-4fae-8770-bdf3e48109ea"
UnknownMemberName="Desconocido">

        </radarcube:TCubeHierarchy>

    </Hierarchies>

</radarcube:TCubeDimension>

    <radarcube:TCubeDimension DisplayName="Sub Sector Economico"
UniqueName="[Sub Sector Economico]">

        <Hierarchies>

            <radarcube:TCubeHierarchy DataTable="SubSectorEconomico"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="SubSectorEconomico"
DisplayFieldType="System.String" DisplayName="Sub Sector Economico"
IDField="SubSectorEconomico" IDFieldType="System.String" Origin="hoAttribute"
TotalCaption="Total" UniqueName="09bf09de-918c-4331-a82c-1ab052f133c9"
UnknownMemberName="Desconocido">

                </radarcube:TCubeHierarchy>

            </Hierarchies>

        </radarcube:TCubeDimension>

        <radarcube:TCubeDimension DisplayName="Tipologia Proyecto"
UniqueName="[Tipologia Proyecto]">

            <Hierarchies>

                <radarcube:TCubeHierarchy DataTable="TipologiaProyecto"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="TipologiaProyecto"
DisplayFieldType="System.String" DisplayName="Tipologia Proyecto"
IDField="TipologiaProyecto" IDFieldType="System.String" Origin="hoAttribute"
TotalCaption="Total" UniqueName="68eb8977-a3a8-4df8-a802-629604209b93"
UnknownMemberName="Desconocido">

                    </radarcube:TCubeHierarchy>

                </Hierarchies>

            </radarcube:TCubeDimension>

            <radarcube:TCubeDimension DisplayName="Departamento"
UniqueName="[Departamento]">

                <Hierarchies>

                    <radarcube:TCubeHierarchy DataTable="Departamento"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Departamento"
DisplayFieldType="System.String" DisplayName="Departamento" IDField="Departamento"

```



```

IDFieldType="System.String" Origin="hoAttribute" TotalCaption="Total"
UniqueName="56990a10-7a07-463a-9ade-ef4d327721e9"
UnknownMemberName="Desconocido">

    </radarcube:TCubeHierarchy>

</Hierarchies>

</radarcube:TCubeDimension>

<radarcube:TCubeDimension DisplayName="anio" UniqueName="[anio]">

    <Hierarchies>

        <radarcube:TCubeHierarchy DataTable="DmFisico"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="Anio"
DisplayFieldType="System.String" DisplayName="Anio" Origin="hoAttribute"
TotalCaption="Total" TypeOfMembers="htNumbers" UniqueName="[anio].[Anio]"
UnknownMemberName="Desconocido">

            </radarcube:TCubeHierarchy>

        </Hierarchies>

    </radarcube:TCubeDimension>

    <radarcube:TCubeDimension DisplayName="tasacambio"
UniqueName="[tasacambio]">

        <Hierarchies>

            <radarcube:TCubeHierarchy DataTable="DmFisico"
DiscretizationBucketFormat="%{First} - %{Last}" DisplayField="TasaCambioUS"
DisplayFieldType="System.String" DisplayName="TasaCambioUS" Origin="hoAttribute"
TotalCaption="Total" TypeOfMembers="htNumbers"
UniqueName="[tasacambio].[TasaCambioUS]" UnknownMemberName="Desconocido">

                </radarcube:TCubeHierarchy>

            </Hierarchies>

        </radarcube:TCubeDimension>

    </Dimensions>

    <CubeTablePositions>

        <radarcube:TCubeTablePosition Name="CatConvenio" Position="233, 15" />

        <radarcube:TCubeTablePosition Name="CatFuente" Position="401, 15" />

        <radarcube:TCubeTablePosition Name="CatInstitucionnivel2" Position="590, 15" />

        <radarcube:TCubeTablePosition Name="CatInstitucionnivel3" Position="779, 15" />

        <radarcube:TCubeTablePosition Name="CatModalidad" Position="912, 15" />

        <radarcube:TCubeTablePosition Name="CatMomento" Position="1045, 15" />

```

```

<radarcube:TCubeTablePosition Name="CatPIP" Position="1204, 15" />
<radarcube:TCubeTablePosition Name="CatRenglones" Position="1359, 15" />
<radarcube:TCubeTablePosition Name="CatSector" Position="1557, 15" />
<radarcube:TCubeTablePosition Name="CatUbicaciones" Position="1736, 15" />
<radarcube:TCubeTablePosition Name="FinRenglones" Position="1882, 15" />
<radarcube:TCubeTablePosition Name="ProyProyecto" Position="2119, 15" />
<radarcube:TCubeTablePosition Name="ActividadObra" Position="578, 15" />
<radarcube:TCubeTablePosition Name="AreaInstitucional" Position="578, 163" />
<radarcube:TCubeTablePosition Name="DmFisico" Position="789, 88" />
<radarcube:TCubeTablePosition Name="Indicador" Position="900, 15" />
<radarcube:TCubeTablePosition Name="Institucion" Position="900, 163" />
<radarcube:TCubeTablePosition Name="Municipio" Position="1047, 15" />
<radarcube:TCubeTablePosition Name="Producto" Position="1047, 163" />
<radarcube:TCubeTablePosition Name="Proyecto" Position="1180, 15" />
<radarcube:TCubeTablePosition Name="SectorEconomico" Position="1180, 205" />
<radarcube:TCubeTablePosition Name="SectorInstitucional" Position="1353, 15" />
<radarcube:TCubeTablePosition Name="SubSectorEconomico" Position="1353, 121"
/>

<radarcube:TCubeTablePosition Name="TipologiaProyecto" Position="1524, 15" />
<radarcube:TCubeTablePosition Name="Departamento" Position="1524, 121" />
</CubeTablePositions>

<TableAdapters>
    <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.ActividadObraTableAdapter"
MethodName="Fill" TableName="ActividadObra" />

    <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.AreaInstitucionalTableAdapter"
MethodName="Fill" TableName="AreaInstitucional" />

    <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.DmFisicoTableAdapter" MethodName="Fill"
TableName="DmFisico" />

    <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.IndicadorTableAdapter" MethodName="Fill"
TableName="Indicador" />

```

```

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.InstitucionTableAdapter" MethodName="Fill"
TableName="Institucion" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.MunicipioTableAdapter" MethodName="Fill"
TableName="Municipio" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.ProductoTableAdapter" MethodName="Fill"
TableName="Producto" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.ProyectoTableAdapter" MethodName="Fill"
TableName="Proyecto" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.SectorEconomicoTableAdapter"
MethodName="Fill" TableName="SectorEconomico" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.SectorInstitucionalTableAdapter"
MethodName="Fill" TableName="SectorInstitucional" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.SubSectorEconomicoTableAdapter"
MethodName="Fill" TableName="SubSectorEconomico" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.TipologiaProyectoTableAdapter"
MethodName="Fill" TableName="TipologiaProyecto" />

        <radarcube:TableAdapterInfo
AdapterType="Pruebaradar.DataSet1TableAdapters.DepartamentoTableAdapter"
MethodName="Fill" TableName="Departamento" />

    </TableAdapters>

</radarcube:TOLAPCube>

```

- **Ejemplo de código para llenar un RadarCube**

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.Sql;

```

```

using System.Data.SqlClient;
using System.Web.Configuration;
using DSBSNIP.DataSet1TableAdapters;

namespace DSBSNIP
{
    public partial class _FISICO : System.Web.UI.Page
    {
        string connectionString=
WebConfigurationManager.ConnectionStrings["SnipDMConnectionString2"].ConnectionString ;
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                TOLAPCube1.Active = false;
            }
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            DataSet1 dm = new DataSet1();

            SqlConnection conexion= new SqlConnection(connectionString);

            // AQUI SE ESTABLECEN LOS ANIOS QUE PRESENTARA EL GRID

            SqlDataAdapter adpaterDMFisico =

                new SqlDataAdapter("SELECT Anio, Cantidad, CodActividadObra,
CodAreaInstitucional, CodDepartamento, CodIndicador, CodInstitucion,
CodMunicipio, CodProducto, CodProyecto, CodSectorEconomico,
CodSectorInstitucional, CodSubSectorEconomico, CodTipologiaProyecto, Costo,
TasaCambioUS FROM DmFisico WHERE (Anio = "+ Convert.ToString(
DropDownList1.SelectedValue)+")", conexion);
            SqlDataAdapter adpaterCodActividadObra =
                new SqlDataAdapter("SELECT CodActividadObra, CodProyecto, CodSnipExt,
ActividadObra FROM dbo.ActividadObra", conexion);
            SqlDataAdapter adpaterCodTipologiaProyecto =
                new SqlDataAdapter("SELECT CodTipologiaProyecto, TipologiaProyecto FROM
dbo.TipologiaProyecto", conexion);
            SqlDataAdapter adpaterCodSubSectorEconomico =
                new SqlDataAdapter("SELECT CodSubSectorEconomico, SubSectorEconomico
FROM dbo.SubSectorEconomico", conexion);
            SqlDataAdapter adpaterCodSectorInstitucional =

```

```

        new SqlDataAdapter("SELECT CodSectorInstitucional, SectorInstitucional FROM
        dbo.SectorInstitucional", conexion);
SqlDataAdapter adpaterCodSectorEconomico =
        new SqlDataAdapter("SELECT CodSectorEconomico, SectorEconomico FROM
        dbo.SectorEconomico", conexion);
SqlDataAdapter adpaterCodProyecto =
        new SqlDataAdapter("SELECT CodProyecto, CodigoSNIP, Proyecto, Siglas,
        FechaInicio, FechaFin FROM dbo.Proyecto", conexion);
SqlDataAdapter adpaterCodProducto =
        new SqlDataAdapter("SELECT CodProducto, Producto FROM dbo.Producto",
        conexion);
SqlDataAdapter adpaterCodMunicipio =
        new SqlDataAdapter("SELECT CodMunicipio, Municipio, CodificadorINIDE,
        EsCabecera FROM dbo.Municipio", conexion);
SqlDataAdapter adpaterCodInstitucion =
        new SqlDataAdapter("SELECT CodInstitucion, Institucion, Siglas FROM
        dbo.Institucion", conexion);
SqlDataAdapter adpaterCodIndicador =
        new SqlDataAdapter("SELECT CodIndicador, Indicador, Magnitud,
        UnidadMedida FROM dbo.Indicador", conexion);
SqlDataAdapter adpaterCodDepartamento =
        new SqlDataAdapter("SELECT CodDepartamento, Departamento,
        CodificadorINIDE FROM dbo.Departamento", conexion);
SqlDataAdapter adpaterCodAreaInstitucional =
        new SqlDataAdapter("SELECT CodAreaInstitucional, AreaInstitucional, Siglas
        FROM dbo.AreaInstitucional", conexion);
conexion.Open();
DataSet1 dsTest= new DataSet1();
adpaterCodActividadObra.Fill(dsTest, "ActividadObra");
adpaterCodTipologiaProyecto.Fill(dsTest, "TipologiaProyecto");
adpaterCodSubSectorEconomico.Fill(dsTest, "SubSectorEconomico");
adpaterCodSectorInstitucional.Fill(dsTest, "SectorInstitucional");
adpaterCodSectorEconomico.Fill(dsTest, "SectorEconomico");
adpaterCodProyecto.Fill(dsTest, "Proyecto");
adpaterCodProducto.Fill(dsTest, "Producto");
adpaterCodMunicipio.Fill(dsTest, "Municipio");
adpaterCodInstitucion.Fill(dsTest, "Institucion");
adpaterCodIndicador.Fill(dsTest, "Indicador");
adpaterCodDepartamento.Fill(dsTest, "Departamento");
adpaterCodAreaInstitucional.Fill(dsTest, "AreaInstitucional");
adpaterDMFisico.Fill(dsTest, "DmFisico");

conexion.Close();

```

```

        adpaterDMFisico.Dispose();
        adpaterCodAreaInstitucional.Dispose();
        adpaterCodDepartamento.Dispose();
        adpaterCodIndicador.Dispose();
        adpaterCodInstitucion.Dispose();
        adpaterCodMunicipio.Dispose();
        adpaterCodProducto.Dispose();
        adpaterCodProyecto.Dispose();
        adpaterCodSectorEconomico.Dispose();
        adpaterCodTipologiaProyecto.Dispose();
        adpaterCodSectorInstitucional.Dispose();
        adpaterCodSubSectorEconomico.Dispose();
        adpaterCodTipologiaProyecto.Dispose();
        adpaterCodActividadObra.Dispose();

        TOLAPGrid1.Enabled = true;
        TOLAPCube1.DataSet = dsTest;
        TOLAPCube1.Active = true;
    }
}
}

```

#### 8.1.4 Procedimiento para generar gráficos con HighCharts

Highcharts es una biblioteca de gráficos escritos en JavaScript puro, que ofrece una manera fácil de añadir gráficos interactivos a un sitio web o aplicación web. Highcharts actualmente suporta gráficos de línea, spline, área, areaspline, columnas, barras, pastel y gráfico de dispersión.

Actualmente existe un Nuget para Visual Studio que permite crear los gráficos de Highcharts en código C# este Nuget se llama Dot.Net Highcharts el cual funciona de interprete entre el servidor y el navegador del usuario creando el código Javascript y renderizando el grafico.

A continuación se establecerán los pasos para crear gráficos con esta librería.

##### Paso 1

Instalar el Nuget de Dot.Net Highcharts a través de la consola de paquetes.

```
PM> Install-Package DotNet.Highcharts
```

## Paso 2.

Añadir la referencia de la librería Javascript en la master page del proyecto

```
<script src="../../Scripts/jquery-1.5.1.min.js" type="text/javascript"></script>  
<script src="../../Scripts/highcharts.js" type="text/javascript"></script>
```

## Paso 3

Una vez añadida la referencia ya podemos empezar a crear los gráficos

Primero en el html de la página que contendrá el grafico, añadimos un **asp:Literal**, en él se renderizará el gráfico

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">  
    <asp:Literal ID="ltrChart" runat="server"></asp:Literal>  
</asp:Content>
```

Seguido esto en el código de la página crearemos el código del chart, cabe resaltar que cada grafica tiene sus propias características.

A continuación se mostrara el código de 2 tipos de Charts, aplicados al presente proyecto donde se muestran los indicadores puntuales de proyectos de inversión pública planificados vs los proyectos ejecutados:

### Ejemplo1 Gauge

```
protected void chartejecucionpip()  
{  
    Highcharts chart = new Highcharts("chartejecucionpip")  
        .InitChart(new Chart{  
            Type = ChartTypes.Gauge,  
            PlotBackgroundColor = null,  
        })  
}
```

```

PlotBackgroundImage = null,
PlotBorderWidth = 0,
PlotShadow = false
})
.SetTitle(new Title { Text = "Porcentaje de ejecucion PIP" })
.SetPane(new Pane{
    StartAngle = -150,
    EndAngle = 150,
    Background = new[]
    {
        new BackgroundObject {
            BackgroundColor =
                new BackColorOrGradient(new Gradient {
                    LinearGradient = new[] { 0, 0, 0, 400 },
                    Stops = new object[,] {
                        {0, Color.FromArgb(255, 96, 96, 96) },
                        {1, Color.FromArgb(255, 16, 16, 16) }
                    }
                })
        },
        BorderWidth = new PercentageOrPixel(0),
        OuterRadius = new PercentageOrPixel(109, true)
    },
    new BackgroundObject {
        BackgroundColor = new BackColorOrGradient(new Gradient
        {
            LinearGradient = new[] { 0, 0, 0, 1},
            Stops = new object[,] { { 0, "#333" }, { 1, "#FFF" } }
        })
        BorderWidth = new PercentageOrPixel(1),
        OuterRadius = new PercentageOrPixel(107, true)
    },
    new BackgroundObject(),
    new BackgroundObject
    {
        BackgroundColor = new BackColorOrGradient(
            ColorTranslator.FromHtml("#DDD")),
        BorderWidth = new PercentageOrPixel(0),
        OuterRadius = new PercentageOrPixel(105, true),
        InnerRadius = new PercentageOrPixel(103, true)
    }
}
})
.SetYAxis(new YAxis
{
    Min = 0,
    Max = 100,

    //MinorTickInterval = "auto",
    MinorTickWidth = 1,

```



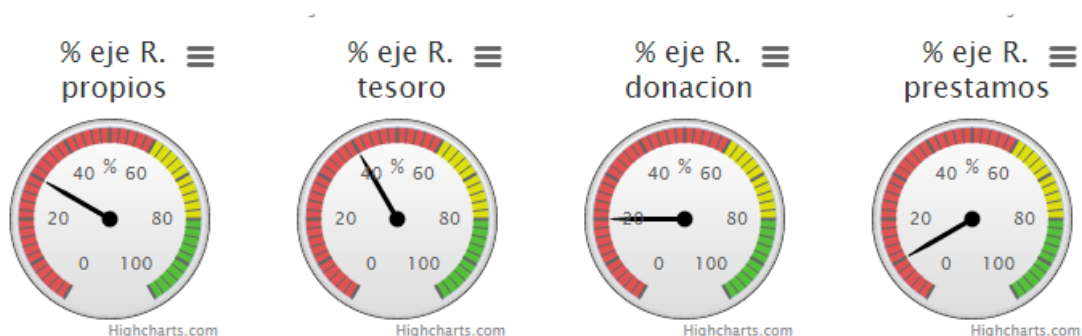
```

MinorTickLength = 10,
MinorTickPosition = TickPositions.Inside,
MinorTickColor = ColorTranslator.FromHtml("#666"),
TickPixelInterval = 30,
TickWidth = 2,
TickPosition = TickPositions.Inside,
TickLength = 10,
TickColor = ColorTranslator.FromHtml("#666"),
Labels = new YAxisLabels
{
    Step = 2,
    //Rotation = "auto"
},
Title = new YAxisTitle { Text = "%" },
PlotBands = new[]
{
    new YAxisPlotBands
        { From = 0, To = 60, Color = ColorTranslator.FromHtml("#DF5353") },
    new YAxisPlotBands
        { From = 60, To = 80, Color = ColorTranslator.FromHtml("#DDDF0D") },
    new YAxisPlotBands
        { From = 80, To = 100, Color = ColorTranslator.FromHtml("#55BF3B") }
}
})
.SetSeries(new Series
{
    Data = new Data(new object[] { valpip })
});
}

ItChart1.Text = chart.ToHtmlString();
}

```

**Ilustración 37: HightChart Gauge**



## Ejemplo2 Gráficos de Columnas

```

protected void chartinstitucion()
{
    Object[] chartValues = new Object[8]; // declare an object for the chart rendering
    DotNet.Highcharts.Highcharts chart = new DotNet.Highcharts.Highcharts("chart")
        .InitChart(new Chart { DefaultSeriesType = ChartTypes.Column })
        .SetTitle(new Title
        {
            Text = "Data",
            X = -20
        })
        .SetSubtitle(new Subtitle
        {
            Text = "Source: Static data",
            X = -20
        })
        .SetXAxis(new XAxis
        {
            Categories = new[] { "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug" }
        })
        .SetSeries(new[]
        {
            new Series
            {
                Name = "# porcentaje por institucion",
                Data = new Data(new object[] { 2, 3, 5, 7, 6, 6, 7, 8 }),
            },
        });
    ltChart2.Text = chart.ToHtmlString();
}

```

### Ilustración 38: Hightchart COLUMNS

